ESIEE ENGINEERING

une école de la Chambre de commerce et d'industrie de Paris

GWA

Rapport

ESIEE I5FR - Projet Fil Rouge 2009 - 2010

Sommaire

Remerciements 4

Introduction 5

I. Présentation du projet 6

I.2) But et enjeu du projet 6

I.1) Qu'est ce que GWA 6

I.3) Contexte 7

I.3.1) Humain 7

I.3.2) Contexte technique 7

I.3.3) Contexte concurrentiel 8

I.4) Bilan de l'existant 10

I.4.1) IHM et fonctions du logiciel 10

I.4.2) Code et base de donnée 11

I.4.3) Documentation 11

I.5) Expression des besoins 12

I.5.1) Fonctions indispensables 12

I.5.2) Fonctions importantes 12

I.5.3) Fonctions supplémentaires 12

I.5.4) Détails des fonctions 13

I.5.5) Contraintes 15

I.6) Périmètre du projet 16

II. Présentation de la solution 17

II.1) Organisation de l'équipe 17

II.2) Présentation des outils utilisés 17

II.2.1) SVN 17

II.2.2) Google Code 21

II.2.3) Environnement de développement 23

II.3) Organigramme des tâches 25

II.4) Description des tâches 26
II.4.1) Prise en main du projet 26
II.4.2) Mise au propre du programme 26
II.4.3) Etude de la géolocalisation 29
II.4.4) Nouvelles fonctionnalités 36
II.4.5) Ergonomie 43
II.4.6) Fiabilisation 44
II.4.7) Distribution 45
II.4.8) Documentation du projet 51
II.5) Diagramme de PERT 55
II.6) Diagramme de GANTT 56
II.7) Indicateurs 57
II.8) Problèmes rencontrés 58
II.8.1) Planification 58
II.8.2) Techniques 58
III. Conclusion 59
III.1) Conclusion technique et humaine 59
III.2) Axes de développement futurs 59
IV. Annexes 61
IV.1) Statistiques 61
IV.2) Manuel d'utilisation 62

Remerciements

Nous tenons tout d'abord à remercier M Olivier Cayrol ainsi que M Sébastien Panel pour l'aide qu'ils nous ont apporté durant la réalisation de ce projet.

Nous remercions également M Daniel Courivaud, notre client, pour sa disponibilité tout au long du projet.

Introduction

Ce projet de fin d'étude (fil-rouge) conclut les trois années d'apprentissage au sein de l'ESIEE. Il s'est déroulé entre Septembre 2009 et Avril 2010 et a représenté un volume de travail d'environ 80 heures.

L'objectif de ce projet était de poursuivre le développement d'un logiciel de mesure et de cartographie de couverture ainsi que de tests de débits Wi-Fi en Python. Nous devions reprendre en main le projet initié l'année précédente par la promotion I5FR 2009 puis continuer son développement afin de l'améliorer et d'y rajouter de nouvelles fonctions.

Nous allons commencer par présenter plus en détail le projet, son contexte, les besoins du client et les problématiques associées. Dans une seconde partie, nous présenterons la solution que nous avons proposée pour répondre aux besoins du client.

I. Présentation du projet

I.2) But et enjeu du projet

Le but du projet de fin d'année d'I5FR, projet « Fil Rouge 2010 » est de continuer le développement du logiciel d'analyse de couverture Wi-Fi, GWA, initié par les étudiants d'I5FR en 2009 à l'ESIEE Paris.

L'enjeu de ce projet est de reprendre en main le développement déjà réalisé, développer de nouvelles fonctions et améliorer le code et la documentation du programme afin que le projet puisse être repris très facilement l'année prochaine par les prochains étudiants et continuer le programme sur des bases saines.

I.1) Qu'est ce que GWA

GWA pour Geographical Wireless Analyser, est un logiciel ayant pour but d'analyser la couverture Wi-Fi d'un bâtiment. L'objectif est de faire une cartographie de la couverture présentant le niveau de puissance reçue en chaque point d'un bâtiment. Cette carte permettra par la suite de pouvoir optimiser la couverture en déplaçant ou en ajoutant des bornes afin d'obtenir la couverture souhaitée.

GWA est un logiciel open source, codé en Python, destiné aux OS de type Linux. Ce logiciel est développé dans le cadre d'un projet de fin d'année pour les étudiants en dernière année d'apprentissage en Réseaux et Télécommunications à l'ESIEE Paris. Ce projet a été initié par M Daniel Courivaud, enseignant et chercheur au laboratoire SIGTEL à l'ESIEE Paris. Une première équipe d'étudiant a démarré le projet en 2009.

I.3) Contexte

I.3.1) Humain

Commanditaire :

- M Daniel Courivaud, enseignant et chercheur au laboratoire SIGTEL à l'ESIEE Paris

Consultants :

- M Olivier Cayrol, consultant, société Logilabs
- M Sébastien Panel, ancien étudiant ayant participé au développement de GWA l'année passé.

L'équipe projet se compose de 7 étudiants d'I5FR promo 2010 :

- David Alexis
- Fabien Bagard
- Fatah Brahmi
- Sébastien Dumoulin
- Sébastien Canchon
- Alexandre Planchon
- Pierre Valat

I.3.2) Contexte technique

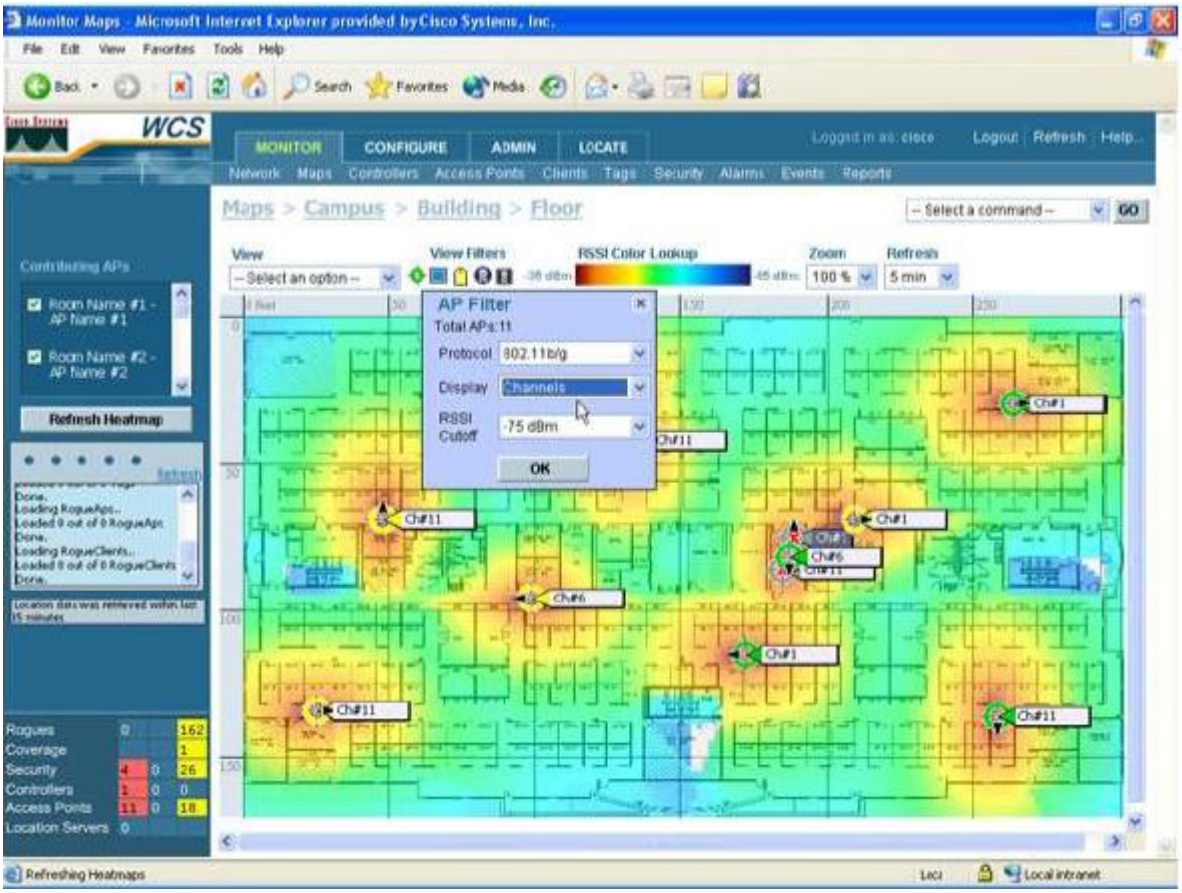
- La solution proposée doit être compatible avec un OS de type Linux.
- Le développement de la solution doit être fait à l'aide de logiciels libres.
- Le développement de la solution doit être fait en utilisant le langage Python

I.3.3) Contexte concurrentiel

D'autres solutions d'analyse et de cartographie de signal Wi-Fi existent sur le marché. Parmi elles on peut retrouver des solutions payantes et d'autres libres. Ci-dessous les 2 solutions les plus populaires :

I.3.3.1) Cisco WCS

WCS, Wireless Control System, est une solution payante fournie par l'équipementier réseau Cisco. Cette solution apporte de très nombreuses fonctions telles que la planification, la configuration, la surveillance et la gestion des réseaux locaux sans fils à partir d'un point unique. En effet, cette solution s'installe sur un serveur à partir duquel elle peut interroger et contrôler tous les AP Wi-Fi et cartographier la couverture radio.



Avantages :

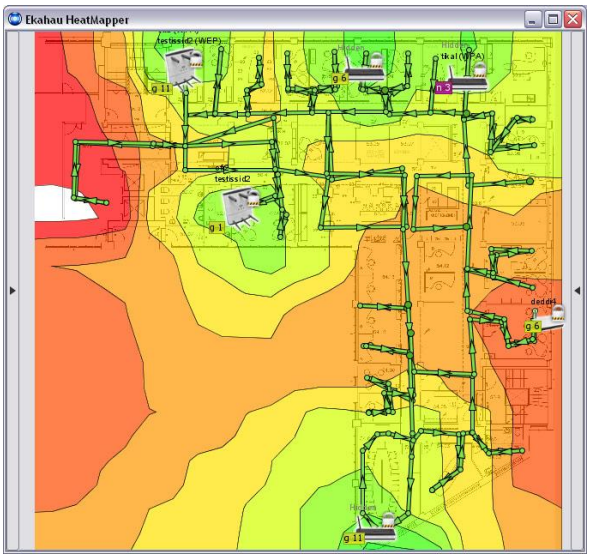
- Produit très complet
- Point unique de gestion
- Outil intégrant des fonctions de reporting
- Support multi OS pour le serveur et les clients (accessible en client web)

Inconvénients :

- Prix de la licence (environ 4000€ pour la gestion de 50 AP maximum)
- Support du protocole LWAPP (Lightweight Access Point Protocol) obligatoire pour les AP.
- Analyse de puissance et cartographie basé sur des protocoles propriétaires
- Nécessite un serveur

I.3.3.2) Ekahau HeatMapper

Ekahau HeatMapper est un outil de cartographie de la couverture pour les réseaux Wi-Fi Il est téléchargeable gratuitement et fonctionne en environnement de Windows. Il est simple d'utilisation et illustre la force des signaux sans fil sur des cartes thermiques avec codes de couleurs. Il permet également de localiser tous les points d'accès audibles et indique leurs configurations et paramètres de sécurité. Il existe également une version payante (Ekahau Site Survey plus avancée)



Avantages :

- Gratuit
- Audit de sécurité des AP

Inconvénients :

- OS Windows uniquement

Au vu de ces logiciels concurrents, on peut en déduire qu'il n'en existe aucun sur le segment visé par GWA c'est-à-dire un outil léger, libre et gratuit et qui fonctionne sous Linux. Ce logiciel dispose donc d'un marché et pourrait devenir un logiciel très populaire dans les années à venir.

I.4) Bilan de l'existant

Notre projet consiste à continuer le travail effectué sur GWA par l'équipe d'étudiants de l'année précédente. Nous avons donc du prendre en main leur travail et faire un bilan pour pouvoir définir les tâches à réaliser.

I.4.1) IHM et fonctions du logiciel

Tâches déjà réalisées :

- Récupération et présentation sur une carte des mesures de puissance
- Récupération des mesures de débits
- IHM du logiciel

Problèmes :

- Résolution de l'IHM trop importante pour l'utilisation sur un ordinateur portable, fenêtres à taille fixe
- Pas de sources des fichiers d'IHM fournies
- Codage directement dans les fichiers Python générés à partir des sources de l'IHM
- Fonction d'export XML non implémentée.

I.4.2) Code et base de donnée

Tâches déjà réalisées :

- Découpage du programme en différents fichiers
- Création de la base de données, des tables et des champs

Problèmes :

- Code non modulaire : découpé en 1 fichier par boite de dialogue
    - Fonctions répétées dans le code de plusieurs pages
    - IHM codée directement en Python et non à partir des sources d'IHM
- Problèmes d'indentation de certaines parties du code
- Pas de fonction de suppression implémentée pour supprimer des éléments de la base.

I.4.3) Documentation

Tâches déjà réalisées :

- Rédaction d'un manuel expliquant le fonctionnement du programme
- Rapport et présentation

Problèmes :

- Code non commenté

I.5) Expression des besoins

I.5.1) Fonctions indispensables

- F1 - Acquisition des mesures d'analyse de puissance Wi-Fi
- F2 - Tests de débits basés sur Iperf
- F3 - Cartographie de couverture
- F4 - Utilisation sur un ordinateur portable avec un OS Linux
- F5 - Stockage des résultats dans une base de données

I.5.2) Fonctions importantes

- F6 - Fonction de translation des mesures d'une salle dans le plan d'un bâtiment
- F7 - Export XML des données
- F8 - Structure modulaire du logiciel
- F9 - Documentation
- F10 - Faciliter la transmission du projet

I.5.3) Fonctions supplémentaires

- F11 - Installation facile
- F12 - Ergonomie du logiciel
- F13 - Clef USB
- F14 - Scan de l'environnement Wi-Fi et audit de sécurité

I.5.4) Détails des fonctions

F1 - Acquisition des mesures d'analyse de puissance Wi-Fi

Le logiciel doit être capable de récupérer le niveau de signal Wi-Fi reçut par la carte pour chaque point de mesure. Pour cela, il doit se baser sur les remontées d'iwlist.

F2 - Tests de débits basés sur Iperf

Le logiciel doit être capable de procéder à un test de débit pour chaque point de mesure basé sur les remontées d'iperf.

F3 - Cartographie de couverture

Le logiciel doit être capable de reporter les mesures de puissance reçue par la carte Wi-Fi pour chaque point de mesure sur une carte de la salle, représentant la couverture Wi-Fi sous forme thermique (de rouge pour un signal très puissant à bleu-violet pour un signal très faible).

F4 - Utilisation sur un ordinateur portable avec un OS Linux

Le logiciel doit pouvoir s'exécuter sur un ordinateur portable équipé d'un OS de type Linux. La résolution d'écran doit donc être adaptée aux tailles d'ordinateurs portables (soit entre 7 et 17 pouces).

F5 - Stockage des résultats dans une base de données

Les résultats de chaque campagne de mesures doivent être sauvegardés au sein d'une base de données afin de pouvoir les récupérer ultérieurement (par exemple pour faire des traitements mathématiques sur les données mesurées).

F6 - Fonction de translation des mesures d'une salle dans le plan d'un bâtiment

Il est important que le logiciel soit capable de replacer les mesures effectuées dans différentes salles d'un bâtiment au sein d'une carte globale pour ce bâtiment.

F7 - Export XML des données

Les résultats d'une campagne doivent pouvoir être exportés au format XML. En effet, ce format standardisé permet de récupérer les données afin de les réutiliser au sein d'autres logiciels. Cela peut permettre par exemple d'exporter les données vers Mathlab afin de réaliser des études mathématiques.

F8 - Structure modulaire du logiciel

Le logiciel doit être conçut de manière modulaire et permette son évolution future de manière simple. Il doit donc reprendre les concepts de la programmation objets.

F9 – Documentation

Il est important que le projet soit bien documenté. Une documentation doit être générée à partir des commentaires insérés au sein même du code. Cette documentation pourra également intégrer un diagramme de classe expliquant graphiquement la relation entre les diverses fonctions. Un manuel d'utilisation doit également être rédigé afin d'expliquer aux utilisateurs la procédure à suivre pour utiliser le logiciel.

F10 - Faciliter la transmission du projet

Le projet ayant pour but de continuer son développement grâce aux équipes d'étudiants des années suivantes, il est nécessaire de tout mettre en œuvre afin de leur faciliter la reprise du projet et l'ajout de nouvelles fonctions. En cela, la fonction de structure modulaire du logiciel ainsi qu'une documentation complète et détaillée permettra d'atteindre ce but.

F11 - Installation facile

Dans la mesure du possible, le logiciel devrait être facilement installable sur une distribution Linux. L'utilisateur devrait pouvoir télécharger et utiliser le logiciel en très peu de temps. Il ne doit pas se préoccuper des éventuelles dépendances du logiciel.

F12 - Ergonomie du logiciel

Le logiciel devra si possible être ergonomique afin de faciliter son utilisation. Elle doit être intuitive et demander le moins possible d'intervention de l'utilisateur.

F13 - Clef USB

Dans la mesure du possible, une clef USB devra être préparée, contenant le logiciel GWA ainsi que tous les autres outils nécessaires pour poursuivre le développement. Celle clef USB devra être bootable et fournit un environnement complet pour l'utilisation de GWA.

F14 - Scan de l'environnement Wi-Fi et audit de sécurité

Une des nouvelles fonctions souhaitées est, si possible, le développement d'un nouveau module permettant de scanner l'environnement Wi-Fi et de pouvoir effectuer un audit de sécurité.

I.5.5) Contraintes

I.7.5.1) Contraintes techniques :

- Environnement Python
- Base de données PostgreSQL
- Interface graphique
- Calculs mathématiques

I.5.5.2) Contrainte de temps :

Le projet doit être réalisé en 80 heures. Des créneaux horaires on été réservés dans l'emploi du temps afin de travailler sur ce projet. La date de fin du projet est fixée pour le 09 Avril 2010.

I.5.5.3) Contrainte de coûts :

Aucun budget n'est prévu pour ce projet. Le développement ainsi que la solution doivent être basée uniquement sur des logiciels libres.

I.6) Périmètre du projet

Le périmètre du projet est le suivant :

- Développement du logiciel à partir de l'existant
- Rédaction d'une documentation complète pour faciliter la reprise par les prochaines équipes

II. Présentation de la solution

II.1) Organisation de l'équipe

GWA étant un logiciel libre, nous nous sommes organisés de la même manière que les équipes de développement de logiciel libre, c'est-à-dire en mode communautaire.

Chacun a pu ainsi travailler sur tous les aspects du projet, sans rester cantonné à une seule partie. Nous avons fait des points d'avancements régulier afin que chacun connaisse l'avancement de chacune des parties, les problèmes rencontrés par les autres, et puisse ainsi y apporter son aide. Ces réunions ont permis également de se répartir les tâches.

II.2) Présentation des outils utilisés

II.2.1) SVN

SVN, signifiant SubVersioN, est un logiciel de gestion de version. Il permet de conserver un historique total du projet et de revenir facilement à une version plus ancienne du programme par exemple.

Nous allons expliquer comment utiliser SVN pour notre projet. Ici seront détaillées les commandes "de base" avec svn, sans interface graphique. Néanmoins, en ayant compris les commandes, l'utilisation via un client graphique tel que TortoiseSVN par exemple ne posera plus aucuns problèmes.

II.2.1.1) Les différents répertoires de SVN

A la racine de SVN se trouvent 3 répertoires de bases:

- **branches**: Ce répertoire héberge les différentes branches d'un même projet. Nous ne l'utiliserons pas.

- **tags**: Ce répertoire héberge les différentes versions du projet. A chaque évolution majeure du code, on copie la version dans ce répertoire. Par exemple, la version initiale du projet a été copiée dans ce répertoire avec le numéro de version 0.1
- **trunk**: ce répertoire contient les sources du projet. Les versions sont gérés via SVN, elles sont appelées 'révision'.

II.2.1.2) Installation

Il faut premièrement s'assurer que SVN est bien installé sur le système:

```
toto@mercure:~$ sudo apt-get install subversion
```

II.2.1.3) Récupération des sources

Avec SVN, la récupération des sources est différente suivant votre contexte. Dans l'ordre, on effectue d'abord un *checkout* puis on effectue des *imports*. Pour récupérer les sources **la première fois**, il faut effectuer un checkout, cela permet de mettre en place l'environnement SVN ainsi que les sources.

```
toto@mercure:~/projet$ ls

manuel.pdf Presentation.pdf technique.pdf

toto@mercure:~/projet$ svn co https://gwa.googlecode.com/svn/trunk/ GWA --username [nom_utilisateur]

A GWA/DOC
A GWA/GWA.sh
A GWA/dumpBD
A GWA/dumpBD/.deleteTable.sql
[...]
A GWA/Projet_final/images/batiment2_mesure.png
A GWA/Projet_final/images/pointeur1.png
Révision 15 extraite.
```

```
toto@mercure:~/projet$ ls
GWA manuel.pdf Presentation.pdf technique.pdf
```

On constate qu'un répertoire GWA a fait son apparition. Ce dernier contient toutes les sources checkouté depuis notre SVN.

```
toto@mercure:~/projet/GWA$ ls -la
total 28
drwxr-xr-x 6 toto toto 4096 2009-09-14 09:38 .
drwxr-xr-x 3 toto toto 4096 2009-09-14 09:38 ..
drwxr-xr-x 3 toto toto 4096 2009-09-14 09:38 DOC
drwxr-xr-x 3 toto toto 4096 2009-09-14 09:38 dumpBD
-rwxr-xr-x 1 toto toto 55 2009-09-14 09:38 GWA.sh
drwxr-xr-x 5 toto toto 4096 2009-09-14 09:38 Projet_final
drwxr-xr-x 6 toto toto 4096 2009-09-14 09:38 .svn
```

Un répertoire '.svn' a aussi fait son apparition: il contient toute la configuration de Subversion (adresse du dépôt, ...) pour notre projet. Maintenant que nous avons une copie en local du dépôt, voyons comment ajouter les UI que nous avons récupérés par exemple.

II.2.1.4) Mise à jour

Dans le jargon SVN, une mise à jour du dépôt se nomme ***commit***. Voyons ici comment rajouter les UI que nous avons récupérés à notre dépôt SVN. Copions le répertoire qui contient les fichiers dans notre répertoire de travail:

```
toto@mercure:~/projet/GWA$ mv ~/pagesui Projet_final/
```

Il faut maintenant signifier à SVN que des nouveaux fichiers ont été ajoutés au projet (étape non nécessaire lors d'une modification d'un fichier existant:

```
toto@mercure:~/projet/GWA$ svn add Projet_final/pagesui
A Projet_final/pagesui
A Projet_final/pagesui/page4.ui
A Projet_final/pagesui/page3.ui
A Projet_final/pagesui/page9.ui
A Projet_final/pagesui/page1.ui
A Projet_final/pagesui/page6.ui
A Projet_final/pagesui/page2.ui
A Projet_final/pagesui/page8.ui
```

```
A Projet_final/pagesui/page5.ui
A Projet_final/pagesui/page7.ui
```

Une étape importante: **avant toute mise à jour du dépôt, toujours faire un update de son répertoire de travail local**

```
toto@mercure:~/projet/GWA$ svn up
À la révision 17.
```

Une fois cette étape effectuée, on peut faire un commit vers le dépôt SVN:

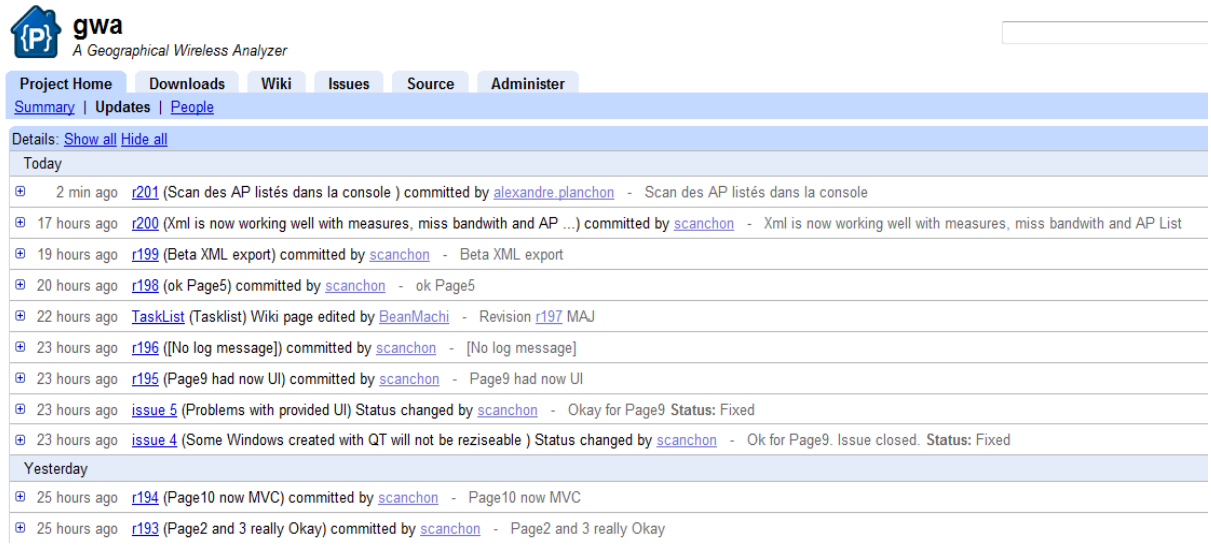
```
toto@mercure:~/projet/GWA$ svn ci -m "un message indiquant ce que vous avez fait"
Ajout Projet_final/pagesui
Ajout Projet_final/pagesui/page1.ui
Ajout Projet_final/pagesui/page2.ui
Ajout Projet_final/pagesui/page3.ui
Ajout Projet_final/pagesui/page4.ui
Ajout Projet_final/pagesui/page5.ui
Ajout Projet_final/pagesui/page6.ui
Ajout Projet_final/pagesui/page7.ui
Ajout Projet_final/pagesui/page8.ui
Ajout Projet_final/pagesui/page9.ui
Transmission des données .....
Révision 18 propagée.
```

Il se peut que SVN vous informe de lignes qui ont été ignorés. Juste fermer le fichier puis taper 'c' pour continuer l'upload. Maintenant, nos fichiers ont été mis à jour sur le dépôt.

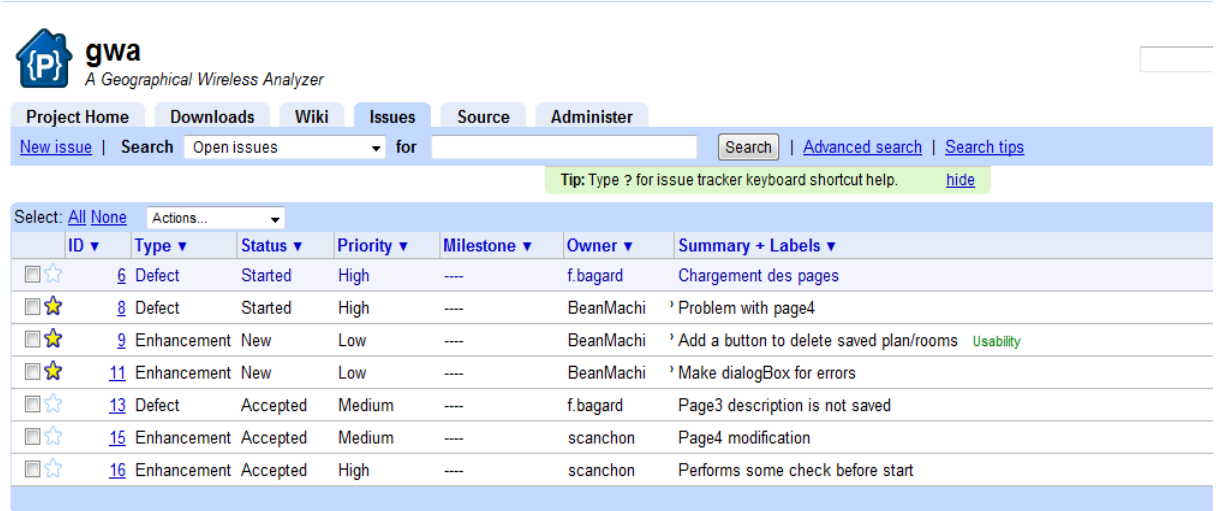
II.2.2) Google Code

Nous avons choisit la plateforme de développement collaboratif de Google, Google Code afin d'héberger notre projet. Ce service est gratuit et permet l'hébergement de projets open source. Il propose diverses fonctionnalités :

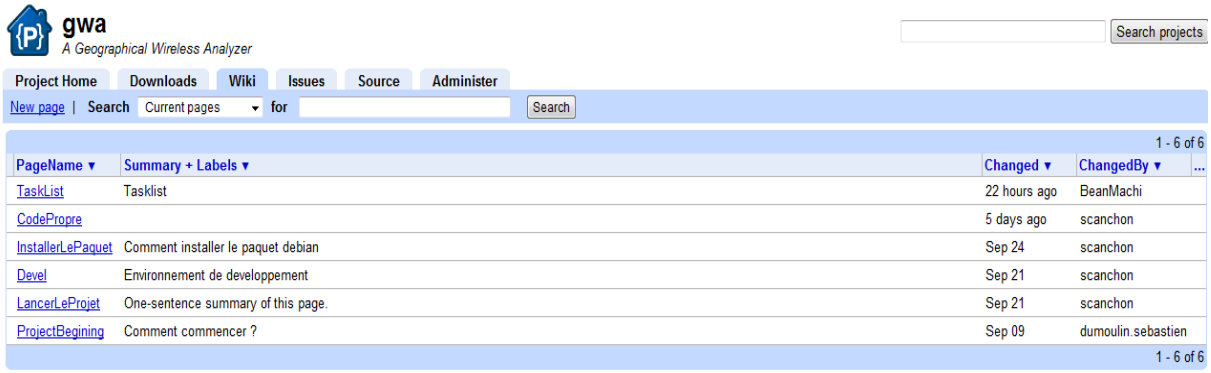
- un gestionnaire de version (SVN ou Mercurial), voir II.1.2 :


gwa
A Geographical Wireless Analyzer
Project Home
Downloads
Wiki
Issues
Source
Administer
Summary | Updates | People
Details: Show all Hide all
Today
2 min ago r201 (Scan des AP listés dans la console) committed by alexandre.planchon - Scan des AP listés dans la console
17 hours ago r200 (Xml is now working well with measures, miss bandwith and AP ...) committed by scanchon - Xml is now working well with measures, miss bandwith and AP List
19 hours ago r199 (Beta XML export) committed by scanchon - Beta XML export
20 hours ago r198 (ok Page5) committed by scanchon - ok Page5
22 hours ago TaskList (Tasklist) Wiki page edited by BeanMachi - Revision r197 MAJ
23 hours ago r196 ([No log message]) committed by scanchon - [No log message]
23 hours ago r195 (Page9 had now UI) committed by scanchon - Page9 had now UI
23 hours ago issue 5 (Problems with provided UI) Status changed by scanchon - Okay for Page9 Status: Fixed
23 hours ago issue 4 (Some Windows created with QT will not be reziseable) Status changed by scanchon - Ok for Page9. Issue closed. Status: Fixed
Yesterday
25 hours ago r194 (Page10 now MVC) committed by scanchon - Page10 now MVC
25 hours ago r193 (Page2 and 3 really Okay) committed by scanchon - Page2 and 3 really Okay

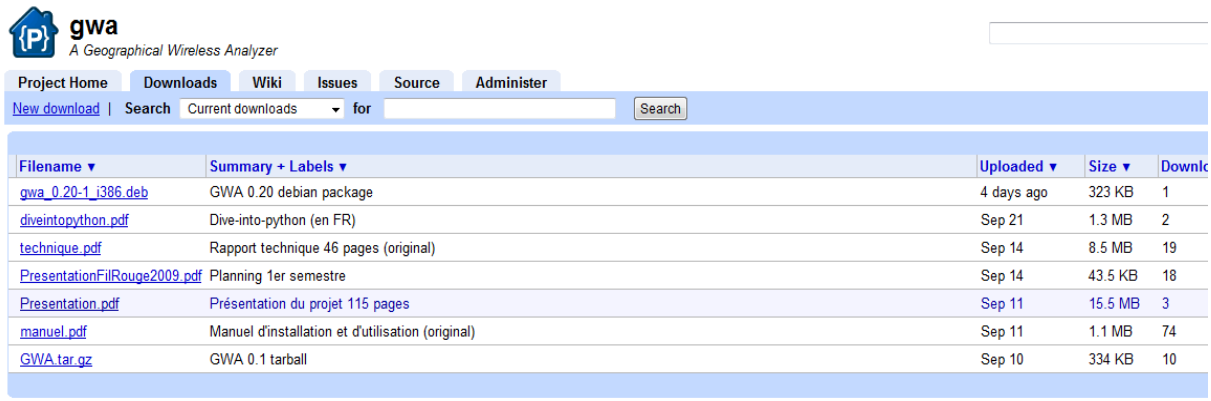
- un gestionnaire de bugs (issues tracker)


gwa
A Geographical Wireless Analyzer
Project Home
Downloads
Wiki
Issues
Source
Administer
New issue | Search Open issues for Search | Advanced search | Search tips
Tip: Type ? for issue tracker keyboard shortcut help. hide
Select: All None Actions...
ID Type Status Priority Milestone Owner Summary + Labels
6 Defect Started High ---- f.bagard Chargement des pages
8 Defect Started High ---- BeanMachi Problem with page4
9 Enhancement New Low ---- BeanMachi Add a button to delete saved plan/rooms Usability
11 Enhancement New Low ---- BeanMachi Make dialogBox for errors
13 Defect Accepted Medium ---- f.bagard Page3 description is not saved
15 Enhancement Accepted Medium ---- scanchon Page4 modification
16 Enhancement Accepted High ---- scanchon Performs some check before start

- un wiki pour la documentation


gwa
A Geographical Wireless Analyzer
Search projects
Project Home
Downloads
Wiki
Issues
Source
Administer
New page | Search Current pages for Search
1 - 6 of 6
PageName ▾ Summary + Labels ▾ Changed ▾ ChangedBy ▾
TaskList Tasklist 22 hours ago BeanMachi
CodePropre 5 days ago scanchon
InstallerLePaquet Comment installer le paquet debian Sep 24 scanchon
Devel Environnement de developpement Sep 21 scanchon
LancerLeProjet One-sentence summary of this page. Sep 21 scanchon
ProjectBegining Comment commencer ? Sep 09 dumoulin.sebastien
1 - 6 of 6

- un espace de téléchargement de fichier

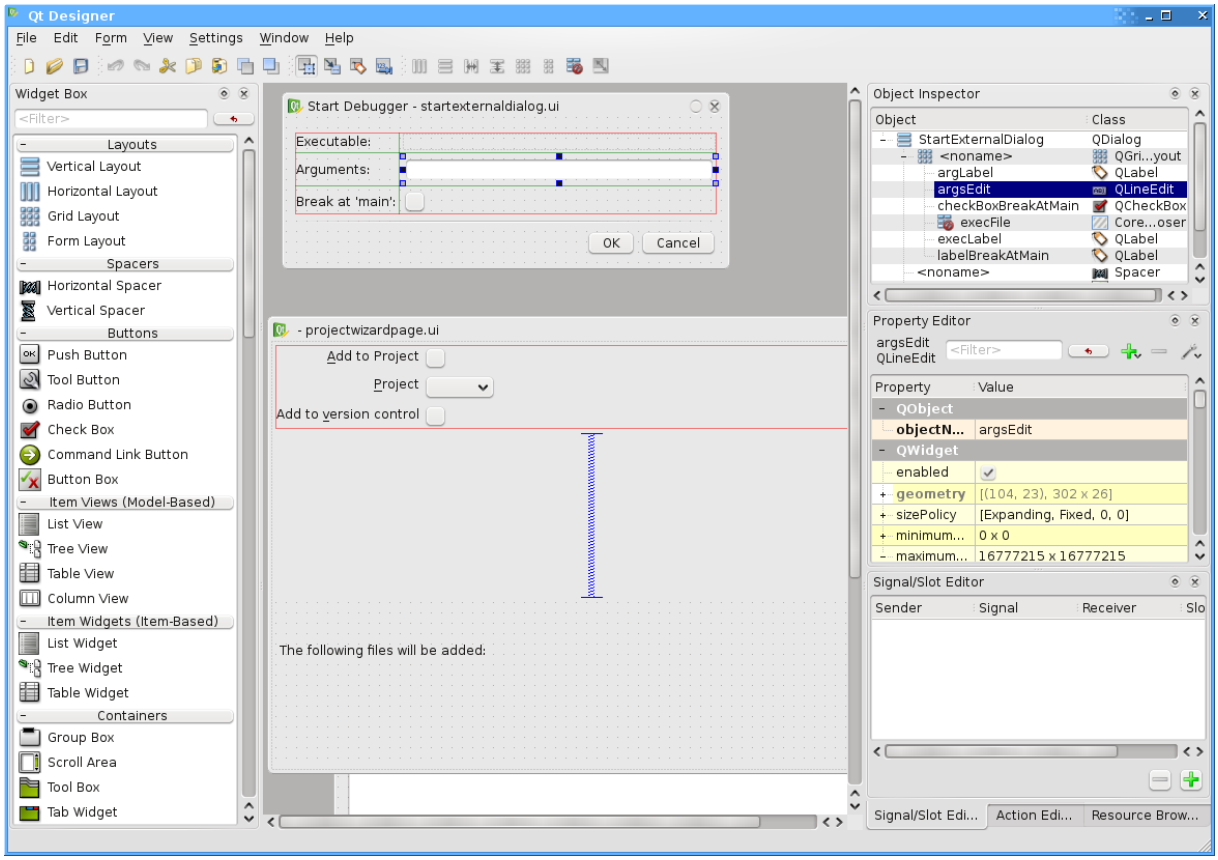

gwa
A Geographical Wireless Analyzer
Project Home
Downloads
Wiki
Issues
Source
Administer
New download | Search Current downloads for Search
Filename ▾ Summary + Labels ▾ Uploaded ▾ Size ▾ Downlo
gwa_0.20-1_i386.deb GWA 0.20 debian package 4 days ago 323 KB 1
diveintopython.pdf Dive-into-python (en FR) Sep 21 1.3 MB 2
technique.pdf Rapport technique 46 pages (original) Sep 14 8.5 MB 19
PresentationFilRouge2009.pdf Planning 1er semestre Sep 14 43.5 KB 18
Presentation.pdf Présentation du projet 115 pages Sep 11 15.5 MB 3
manuel.pdf Manuel d'installation et d'utilisation (original) Sep 11 1.1 MB 74
GWA.tar.gz GWA 0.1 tarball Sep 10 334 KB 10

Le site est accessible à l'adresse suivante : http://code.google.com/p/gwa/

II.2.3) Environnement de développement

II.2.3.1) QT Designer 4

QT Designer est un logiciel permettant la création d'interfaces graphiques (IHM) de manière simple. L'interface du logiciel permet de créer les fenêtres graphiques en mode WYSIWYG. Il est basé sur l'utilisation de la bibliothèque graphique Qt, développée en C++. Cet outil génère des fichiers .ui (User Interface) contenant la description de toute la partie graphique en XML.



II.2.3.2) PyQT4

PyQT4 est un programme permettant de convertir les fichiers .ui en fichier Python .py. Il converti donc en fait le code XML en code Python. Pour cela, il suffit de d'entrer la commande suivante :

```
pyuic4 -o [fichier.py] [fichier.ui]
```

II.2.3.3) Editeur

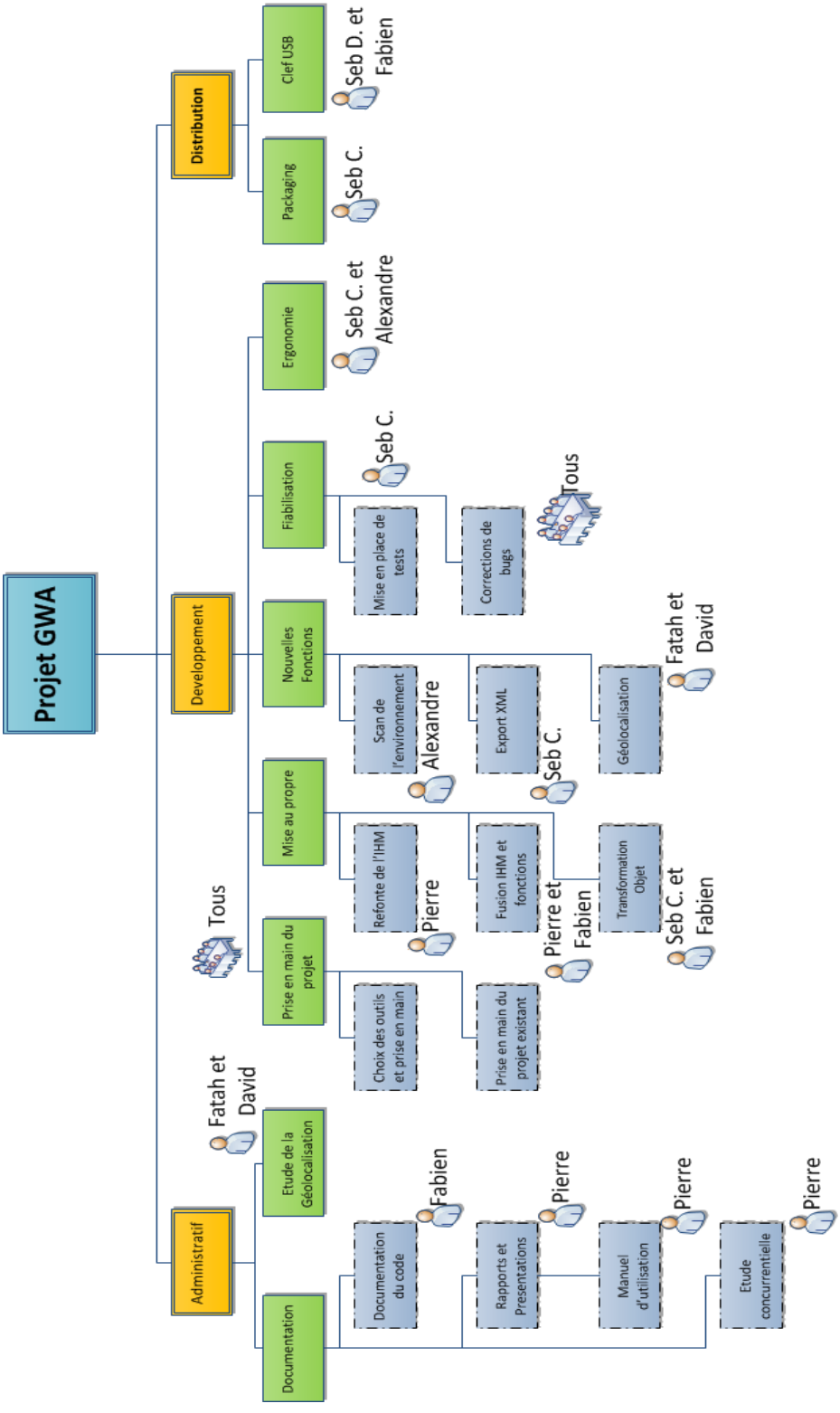
Il existe de nombreux éditeurs permettant de reconnaitre le langage Python. Il existe également un IDE Python complet, Eric.

Nous avons choisit de ne pas imposer d'éditeur, chacun travaillant avec celui avec lequel il est le plus à l'aise. Au final personne n'a utilisé l'IDE Eric car trop lourd et complexe pour notre usage.

Exemple d'éditeurs utilisés :

- Kate
- gedit
- Vi

II.3) Organigramme des tâches



II.4) Description des tâches

II.4.1) Prise en main du projet

Le projet étant la reprise du projet GWA de l'année précédente, il était nécessaire pour tout le groupe projet d'étudier en détail ce qui avait été réalisé par l'équipe précédente. Pour cela, nous avons étudié la documentation fournie ainsi que le code afin de s'approprier le projet et bien comprendre le fonctionnement des différentes classes et fonctions, la structure de la base de données, etc.

Nous avons également tenté d'utiliser le logiciel afin de tester son fonctionnement. Cependant, de nombreuses erreurs lors du lancement ainsi qu'une résolution beaucoup trop élevée pour l'utilisation sur un ordinateur portable nous ont obligés à d'abord passer par une grande étape de mise au propre du programme.

II.4.2) Mise au propre du programme

II.4.2.1) Refonte de l'IHM

L'Interface Homme Machine (IHM) crée par l'équipe précédente n'était pas redimensionnable. La résolution nécessaire pour pouvoir afficher le programme devait être supérieure à 1 181×884, ce qui n'est pas supporté sur les ordinateurs portables (nécessitent un écran de 19 pouces ou plus).

L'IHM avait été crée par l'équipe précédente à l'aide du logiciel QT4 designer. Ce logiciel crée des fichiers d'IHM en .ui qui sont ensuite compilé à l'aide de PyQT4 en .py (Python).

Les fichiers sources d'IHM n'étant pas inclus dans les sources du programme, nous avons donc du les demander à l'ancienne équipe. Les fichiers .ui qu'ils nous ont fournis ne correspondaient plus à l'IHM actuelle. En effet, ils avaient modifié directement les fichiers Python générés à partir des .ui.

Afin de rester modulaire et de pouvoir modifier l'IHM facilement, nous avons choisi de reprendre l'IHM de zéro et de recréer les .ui.

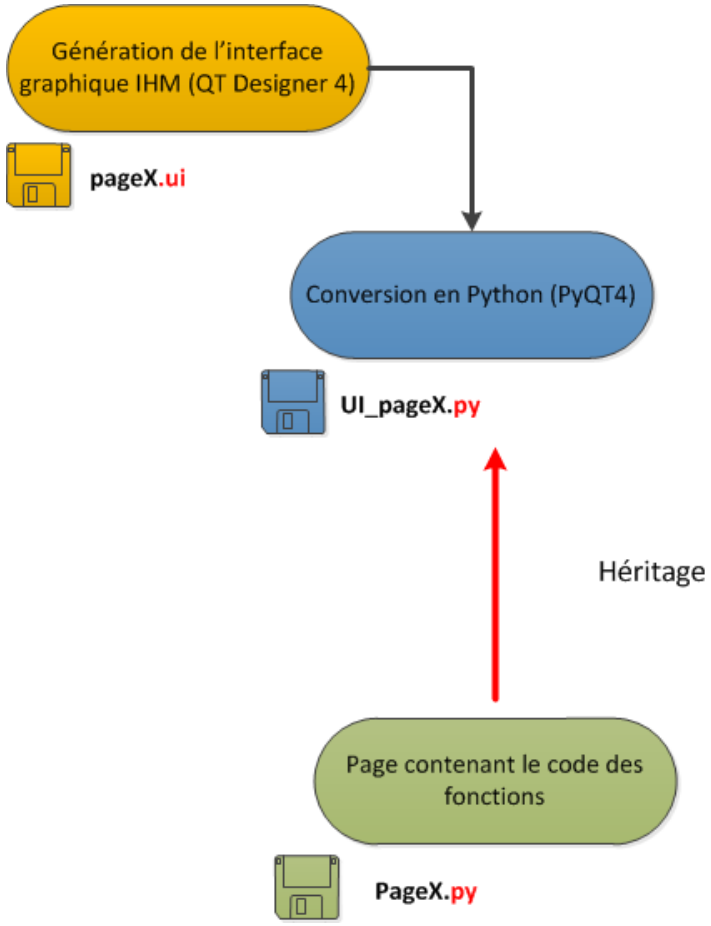
Cela nous a permis d'avoir des fenêtres d'une résolution de 1024x780, résolution supportée par les ordinateurs portables et les notebooks. De plus, nous avons ajouté la possibilité de redimensionner les fenêtres afin de pouvoir adapter facilement la taille des boites de dialogues à la résolution d'écran utilisée.

II.4.2.2) Fusion IHM et fonctions

Une fois l'IHM recrée, nous devions réintégrer le code du programme dans les pages Python générées par PyQT4. En effet, l'équipe précédente avait codé les fonctions au sein même des pages de l'IHM. Il a donc fallu pour chacune des pages, identifier quel était le code Python correspondant à l'IHM et quel était le code Python correspondant au fonctionnement du programme. De plus, l'IHM ayant changée, il nous a fallu adapter l'ancien code à la nouvelle IHM. Nous n'avions pas encore eu les cours sur le langage Python. Cette étape a donc été assez longue et fastidieuse.

II.4.2.3) Transformation objet

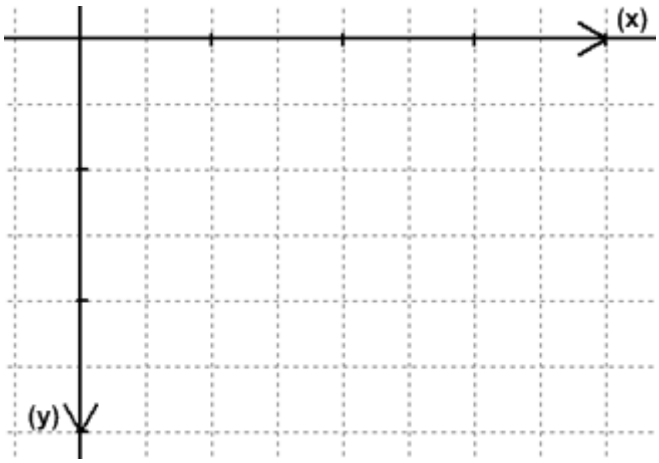
Une fois l'IHM recréé et le programme fonctionnel, nous avons choisit de transformer le découpage du logiciel afin de le rendre modulaire. En effet, au vu du temps passé pour fusionner le code du programme avec celui de l'IHM, nous avons décidé de les séparer afin de pouvoir régénérer une nouvelle IHM à tout moment sans impacter le fonctionnement du programme. Pour cela, nous avons transformé le code en code objet afin de gagner en souplesse. Cela nous a également permit d'utiliser la relation d'héritage.



II.4.3) Etude de la géolocalisation

II.4.3.1) Contexte :

Actuellement, dans le programme récupéré à la promotion I5FR de l'an dernier, un repère global a été définit. Ce repère, global au bâtiment, est utilisé pour effectuer les calculs.Le repère global au bâtiment est définit en haut à gauche du bâtiment de la manière suivante :



Les coordonnées du point d'origine du repère local fourni par l'utilisateur du programme est défini par rapport aux coordonnées du point d'origine du repère global au bâtiment. Un problème subsiste avec le repère global définit par la promotion de l'an dernier. En effet, les axes des abscisses et des ordonnées (x et y) sont mal orientés. L'orientation de ces axes ne respecte pas la « règle de la main droite ».

La règle de la main droite est un moyen de se rappeler comment sont liées diverses directions. Avec cette règle, nous pouvons représenter facilement un repère orthonormal direct. Le pouce, l'index et le majeur permettent de représenter les trois vecteurs de la base appelée couramment (i,j,k) ou encore (x,y,z). Les trois doigts forment alors un trièdre dans l'espace.

pouce = $\vec{x}$
index = $\vec{y}$
majeur = $\vec{z}$

L'orientation choisie des axes ne respecte pas la règle de la main droite et complexifie donc les calculs à réaliser pour comparer les valeurs des différentes mesures effectuées par notre programme.

II.4.3.2) Solution proposée

Nous voulons dans un premier temps redéfinir le repère global afin que les axes de ce repère respectent la règle de la main droite. Nous allons donc définir un repère en orientant l'axe des abscisses vers la droite et celui des ordonnées vers le haut.

A présent, nous souhaitons trouver un moyen de rapporter les coordonnées d'un repère local au repère global afin de pouvoir comparer les différentes mesures effectuées par notre programme. Pour ce faire nous avons décidé d'utiliser plusieurs opérations : une translation et une rotation.

De façon générale, une translation sert à déplacer un objet dans l'espace sans le tourner. Dans notre cas, nous utiliserons un vecteur de translation afin de positionner le point d'origine des différents repères locaux aux coordonnées du point d'origine global au bâtiment.

Les matrices de translation sont utilisées pour tourner un ensemble de points dans un système de coordonnées. Chaque point possède de nouvelles coordonnées, mais leurs distances relatives restent inchangées. Dans notre cas, une matrice de rotation sera utilisée afin d'orienter les axes des repères locaux dans la même direction et le même sens que les axes du repère global.

Pour simplifier nos calculs, le repère global reste toujours le même, quelque soit le bâtiment, c'est-à-dire que l'orientation des axes de ce repère sera inchangée et l'origine du repère global aura pour coordonnées (0, 0, 0). De plus, l'orientation de l'axe z ne sera jamais modifiée pour les repères locaux et sera donc identique à celle du repère global.

Le fait de ne pas modifier l'orientation de l'axe z simplifie nos calculs et nous permet d'effectuer nos opérations en 2 dimensions.

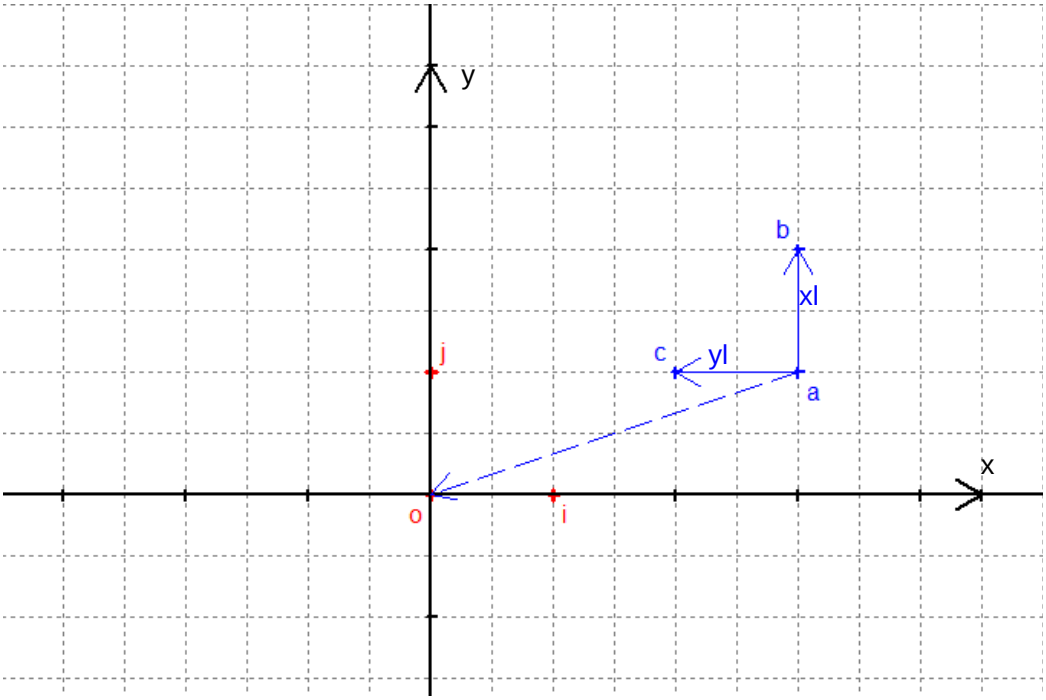
Enfin, nous utiliserons la même échelle dans nos différents repères, ceci nous évitera d'avoir à utiliser une matrice de mise à l'échelle.

II.4.3.3) Opérations

Translation :

Un vecteur de translation est utilisé pour ramener le point d'origine d'un repère local aux coordonnées du point d'origine du repère global.

Prenons un exemple. Le graphique suivant présente un repère orthonormé global en noir dont l'origine est le point « o », un autre repère dont l'origine est « a » est représenté en bleu. Ici, nous voulons ramener le point « a » aux coordonnées du point « o ».



Nous avons les coordonnées suivantes :

$a\begin{pmatrix}3\\1\end{pmatrix}$; $b\begin{pmatrix}3\\2\end{pmatrix}$; $c\begin{pmatrix}2\\1\end{pmatrix}$ et nous recherchons les composantes du vecteur $\vec{T}\begin{pmatrix}m\\n\end{pmatrix}$

Nous cherchons à appliquer au point « a » un vecteur de translation dont les composantes sont déterminées à l'aide des équations suivantes :

$$a\begin{pmatrix}3\\1\end{pmatrix}+\vec{T}\begin{pmatrix}m\\n\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix}$$

Ceci revient à résoudre

$(3+m=0)$
$(1+n=0)$

Le vecteur $\vec{T}$ a pour composante $\vec{T}\begin{pmatrix}-3\\-1\end{pmatrix}$

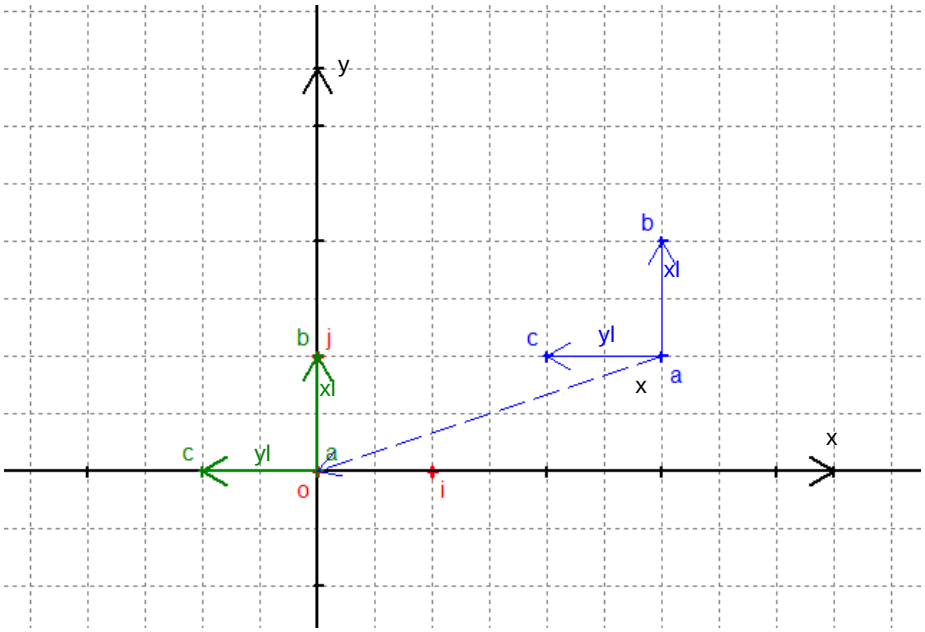
De manière générale, pour tout point « a » (origine d'un repère local) de composante $\begin{pmatrix}x\\y\end{pmatrix}$

nous appliquerons un vecteur de translation $\vec{T}$ de la forme suivante :

$\vec{T}\begin{pmatrix}-x\\-y\end{pmatrix}$

Ce vecteur est à appliquer aux différents points appartenant au repère local en question.

En appliquant le vecteur de translation aux points a, b et c nous avons le résultat suivant (en vert) :



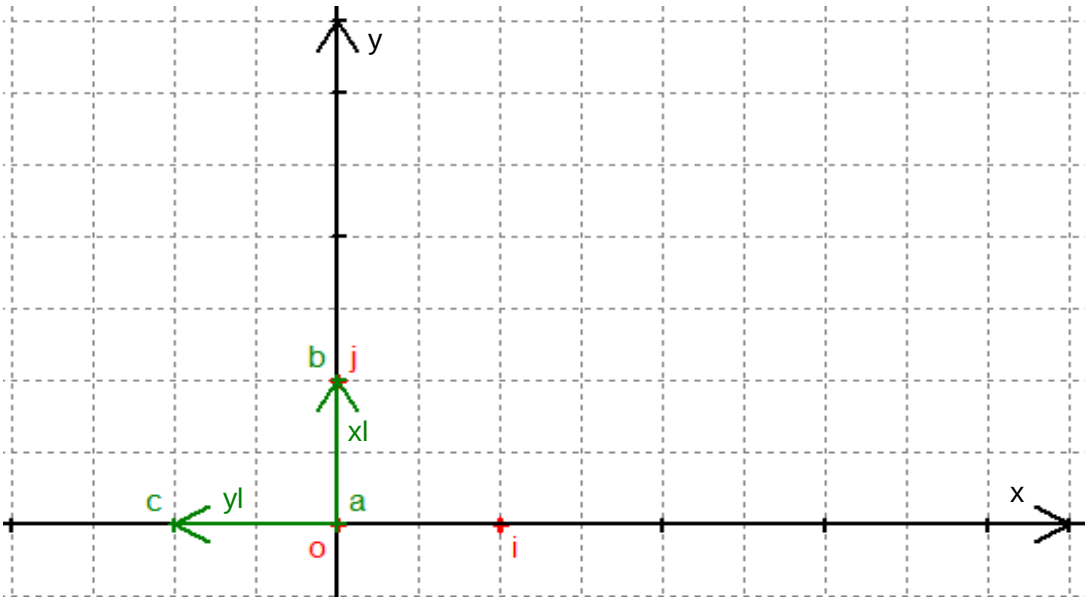
Rotation :

Comme expliqué plus haut, la matrice de rotation a pour but de tourner un ensemble de points dans un système de coordonnées.

Effectuer une rotation à un point revient à multiplier ses coordonnées par une matrice de rotation, nous allons utiliser une matrice de rotation sur 2 dimensions :

$$r(\theta)=\begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}$$

Si l'on reprend l'exemple précédent, nous avons le graphique suivant :



Nos points a, b et c ont pour coordonnées :

$$a\begin{pmatrix}0\\0\end{pmatrix};\ b\begin{pmatrix}0\\1\end{pmatrix};\ c\begin{pmatrix}-1\\0\end{pmatrix}$$

$$r(\theta)=\begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}$$

Ici, $\theta=\frac{3\pi}{2}$. Cet angle nous permet de faire concorder les axes du repère local (en vert) au repère global (en noir).

Nous appliquons la matrice de rotation aux différents points du repère local de la manière suivante :

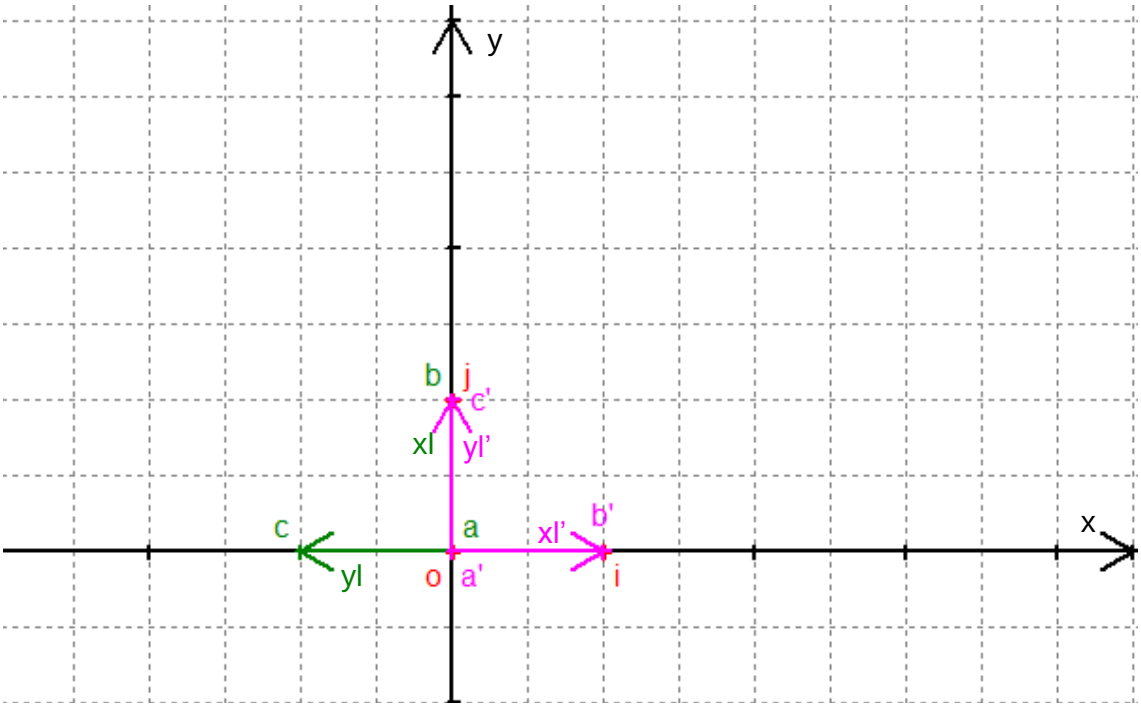
X'=R*X avec $R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$, X représente les coordonnées de l'ancien point et X' les coordonnées du nouveau point.

$$a' = R * a = \begin{pmatrix} \cos\frac{3\pi}{2} & -\sin\frac{3\pi}{2} \\ \sin\frac{3\pi}{2} & \cos\frac{3\pi}{2} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$b' = R * b = \begin{pmatrix} \cos\frac{3\pi}{2} & -\sin\frac{3\pi}{2} \\ \sin\frac{3\pi}{2} & \cos\frac{3\pi}{2} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\sin\frac{3\pi}{2} \\ \cos\frac{3\pi}{2} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$c' = R * c = \begin{pmatrix} \cos\frac{3\pi}{2} & -\sin\frac{3\pi}{2} \\ \sin\frac{3\pi}{2} & \cos\frac{3\pi}{2} \end{pmatrix} \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -\cos\frac{3\pi}{2} \\ -\sin\frac{3\pi}{2} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

En représentant nos résultats sur le graphique (en rose), nous voyons que le nouveau repère obtenu est bien en concordance avec notre repère global. Cette matrice doit être appliquée à tous les points appartenant au repère local.



Le repère local est maintenant en concordance avec notre repère global. En réalisant les mêmes opérations pour nos différents repères locaux, nous pourrons comparer les différentes valeurs obtenues à l'aide de notre programme.

II.4.4) Nouvelles fonctionnalités

II.4.4.1) Géolocalisation

Pour implémenter les différentes opérations relatives à la géolocalisation, nous avons utilisé « numpy ». Numpy est une extension du langage de programmation Python, destinée à manipuler des matrices ou des tableaux multidimensionnels. Cette extension va donc nous permettre de réaliser nos différents calculs, à savoir la translation et la rotation.

Nous avons donc définit une méthode « changementRepere » qui a pour but de calculer et retourner les nouvelles coordonnées d'un point appartenant à un repère local afin que ce point appartiennent au repère global :

```
changementRepere(xorigine,yorigine,xlocal,ylocal,teta=0)
```

Cette méthode prend en compte plusieurs paramètres. Les paramètres « xorigine » et « yorigine » représentent les coordonnées du point d'origine du repère local. C'est à partir de ces coordonnées que l'on déterminera les composantes de notre vecteur de translation. Ensuite, les paramètres « xlocal » et « ylocal » représentent les coordonnées d'un point appartenant au repère local, notre méthode va définir les nouvelles coordonnées de ce point dans le repère global. Enfin, le dernier paramètre, « teta », représente l'angle entre l'axe des abscisses du repère global et l'axe des abscisses du repère local. Cet angle est utilisé pour effectuer la rotation.

II.4.4.2) Scan de l'environnement

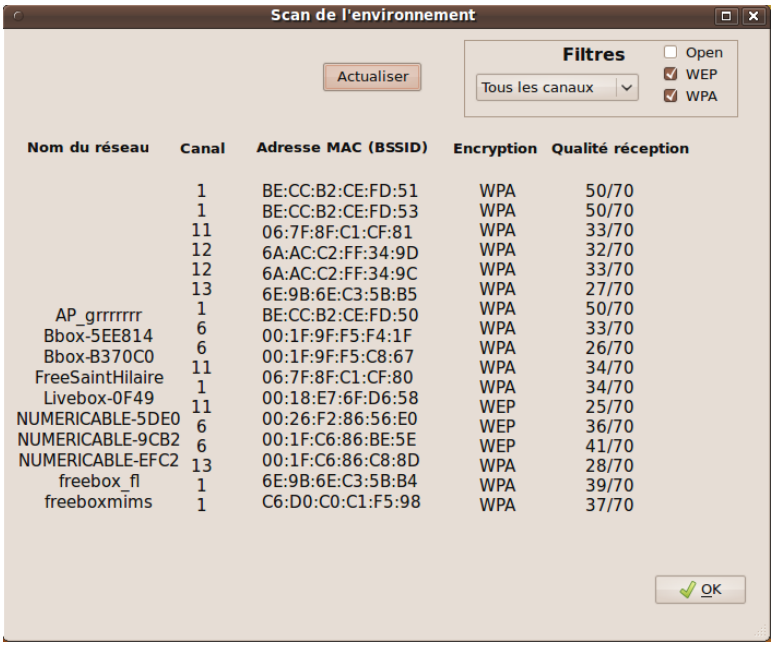
Une des fonctions optionnelles demandées par le client était d'implémenter un module de scan de l'environnement et de sécurité Wi-Fi. Nous avons ainsi décidé d'implémenter la fonction de scan de l'environnement. Celle-ci permet de connaître l'environnement radio dans lequel on se situe et voir quelle sécurité est appliquée (Open, WEP, WPA,..).

Cette nouvelle fonctionnalité correspond à une nouvelle page dans l'IHM, accessible au démarrage de l'application. En utilisant les résultats de la commande iwlist, les données sont représentées sous la forme d'un tableau dans lequel chaque ligne correspond à un point d'accès.

Ce tableau permet de connaître pour chaque Access point :

- Le nom du réseau (ESSID) --> sera vide si le réseau ne diffuse pas son nom
- Le canal --> Tous les canaux supportés par la carte réseau seront parcourus
- L'adresse MAC (BSSID)
- Le type de cryptage utilisé --> WEP, WPA ou pas de cryptage (open)
- La qualité de la réception

Chaque champs du tableau a été rendu sélectionnable pour permettre le copier/coller.



Les filtres :

Une section permettant de filtrer les résultats a également été implémentée. Celle-ci permet de n'afficher que certains AP en utilisant des critères de cryptage ou bien de se restreindre à un canal en particulier.

Ce filtre s'avère très utile lorsqu'un grand nombre de points d'accès sont captés à proximité (typiquement dans les immeubles en ville), et l'on peut rapidement se retrouver avec plus d'une vingtaine d'AP.

La plupart des cartes wifi permettent d'utiliser les fréquences du standard WiFi 802.11a (bande 5Ghz). Le filtre prend donc en compte en plus des 13 canaux du 802.11b/g, les canaux autorisés en Europe du 802.11a. En effet, la réglementation en matière de fréquences radio diffère selon les pays (cf. tableau ci-dessous).

Fréquences dans la bande 2,4 Ghz			**Fréquences dans la bande 5 Ghz (802.11a)**				
Canal	**Fréquence (Mhz)**	**Pays**	**Canal**	**Fréquence (Mhz)**	**USA**	**Europe**	**Japon**
1	2412	Japon, Europe ETSI, États-Unis FCC	34	5170	No	No	Yes
			36	5180	Yes	Yes	Yes
2	2417	Japon, Europe ETSI, États-Unis FCC	38	5190	No	No	Yes
3	2422	Japon, Europe ETSI, États-Unis FCC	40	5200	Yes	Yes	Yes
			42	5210	No	No	Yes
4	2427	Japon, Europe ETSI, États-Unis FCC	44	5220	Yes	Yes	Yes
5	2432	Japon, Europe ETSI, États-Unis FCC	46	5230	No	No	Yes
			48	5240	Yes	Yes	No
6	2437	Japon, Europe ETSI, États-Unis FCC	52	5260	Yes	Yes	No
7	2442	Japon, Europe ETSI, États-Unis FCC	56	5280	Yes	Yes	No
			60	5300	Yes	Yes	No
8	2447	Japon, Europe ETSI, États-Unis FCC	64	5320	Yes	Yes	No
9	2452	Japon, Europe ETSI, États-Unis FCC	100	5500	No	Yes	No
			104	5520	No	Yes	No
10	2457	Japon, Europe ETSI, États-Unis FCC	108	5540	No	Yes	No
			112	5560	No	Yes	No
11	2462	Japon, Europe ETSI,					

12	2467	États-Unis FCC Japon, Europe ETSI	116	5580	No	Yes	No
			120	5600	No	Yes	No
			124	5620	No	Yes	Yes
13	2472	Japon, Europe ETSI	128	5640	No	Yes	Yes
			132	5660	No	Yes	Yes
14	2484	Japon uniquement	136	5680	No	Yes	No
			140	5700	No	Yes	Yes
			149	5745	Yes	No	No
			153	5765	Yes	No	No
			157	5785	Yes	No	No
			161	5805	Yes	No	No
			165	5825	Yes	No	No

Remarque :

En pratique, le balayage des cellules à proximité se fait plus rapidement si la station n'est connectée à aucun AP. C'est pourquoi le programme regarde si la station est connectée et, le cas échéant, affiche un message qui conseille de se déconnecter.

II.4.4.3) Export XML

GWA permet d'exporter a la fin d'une campagne toutes les informations collectées. Le format d'export est XML (eXtensible Markup Language). C'est un langage de balisage générique. L'avantage de ce format est qu'il sera simple de l'importer dans d'autres logiciels (pour traiter plus finement les résultats par exemple).

Voici un exemple volontairement raccourci d'un fichier exporté par GWA :

```
<Resultats>
 <Batiment>
  <id>2010-3-17-16-31-17-274988</id>
  <nom>beaubatiment</nom>
  <x>0</x>
  <y>0</y>
  <realwidth> 1000.000000</realwidth>
  <realheight> 1000.000000</realheight>
 </Batiment>
 <Salle>
  <id>2010-3-17-21-53-13-215040</id>
  <nom>2400</nom>
  <description />
  <x>0</x>
  <y>0</y>
  <z>0</z>
  <realwidth> 210.000000</realwidth>
  <realheight> 750.000000</realheight>
 </Salle>
 <Campagne>
  <nom>zsedrftg</nom>
  <statradio>1</statradio>
  <automatique>1</automatique>
  <step>50</step>
  <env>0</env>
  <manuel>0</manuel>
  <stattrafic>0</stattrafic>
  <id>2010-4-6-16-33-8-364284</id>
 </Campagne>
```

```
<Mesures>
 <2010-4-6-16-33-20-400367>
  <x>50</x>
  <y>50</y>
  <z>0</z>
  <description>z</description>
  <Link Quality>76/100</Link Quality>
  <Rx invalid frag>0.0</Rx invalid frag>
  <Power Management>off</Power Management>
  <Fragment thr />
  <Invalid misc>0.0</Invalid misc>
  <802.11>b</802.11>
  <Tx-Power>27 dBm</Tx-Power>
  <Tx excessive retries>0.0</Tx excessive retries>
  <Missed beacon>0.0</Missed beacon>
  <Bit Rate>48</Bit Rate>
  <Rx invalid crypt>0.0</Rx invalid crypt>
  <Encryption key />
  <Signal level>-54 dBm</Signal level>
  <ESSID>"FIRES"</ESSID>
  <Access Point>00:1D:60:15:60:EE</Access Point>
  <Noise level>-103 dBm</Noise level>
  <Rx invalid nwid>0.0</Rx invalid nwid>
  <Sensitivity />
  <RTS thr>off</RTS thr>
  <Frequency>2.422 GHz</Frequency>
  <Retry limit>None</Retry limit>
  <Mode>Managed</Mode>
 </2010-4-6-16-33-20-400367>
 <2010-4-6-16-33-26-817246>
  <x>50</x>
  <y>100</y>
  <z>0</z>
  <description>z</description>
  <Link Quality>76/100</Link Quality>
  <Rx invalid frag>0.0</Rx invalid frag>
  <Power Management>off</Power Management>
  <Fragment thr />
```

```
    <Invalid misc>0.0</Invalid misc>
    <802.11>b</802.11>
    <Tx-Power>27 dBm</Tx-Power>
    <Tx excessive retries>0.0</Tx excessive retries>
    <Missed beacon>0.0</Missed beacon>
    <Bit Rate>48</Bit Rate>
    <Rx invalid crypt>0.0</Rx invalid crypt>
    <Encryption key />
    <Signal level>-55 dBm</Signal level>
    <ESSID>"FIRES"</ESSID>
    <Access Point>00:1D:60:15:60:EE</Access Point>
    <Noise level>-104 dBm</Noise level>
    <Rx invalid nwid>0.0</Rx invalid nwid>
    <Sensitivity />
    <RTS thr>off</RTS thr>
    <Frequency>2.422 GHz</Frequency>
    <Retry limit>None</Retry limit>
    <Mode>Managed</Mode>
  </2010-4-6-16-33-26-817246>
 </Mesures>
</Resultats>
```

La syntaxe XML a été volontairement faite pour qu'elle soit interprétable par un homme. L'export commence par détailler toutes les informations qu'il a pour le bâtiment en cours (balise *<batiment />*), vient ensuite les informations sur la salle (balise *<salle />*), puis le détail de la campagne, notamment quels ont été les tests effectués (balise *<campagne />*).

Enfin pour finir l'export de toutes les mesures (dans le cas de l'exemple, uniquement la partie radio) dans la balise *<mesure />*. Chaque mesure est séparée par une balise qui contient son identifiant unique de mesure. On retrouvera dedans les coordonnées relatives de la mesure par rapport à la salle, les informations radios et/ou débit, etc …

Nous n'avons pas jugé nécessaire d'exporter les infos sur l'AP car on retrouve ces dernières sur la partie radio des mesures.

II.4.5) Ergonomie

Afin d'améliorer l'ergonomie du logiciel, nous avons travaillé tout d'abord sur l'IHM. Nous l'avons tout d'abord rendue redimensionnable dynamiquement. Cela permet d'adapter l'interface à la résolution d'écran du pc portable. Nous avons ensuite choisit de simplifier cette interface en supprimant les champs non nécessaire et en replaçant les boutons de manière plus intuitive. Nous avons également implémenté des raccourcis claviers afin de lancer les mesures. Cela permet d'éviter à l'utilisateur de devoir faire les mesures avec une souris ou en touchant en permanence le touchpad (ce qui n'est pas évident quand il se trouve debout avec le pc dans les bras). Il lui suffit simplement d'appuyer sur une touche pour passer à la mesure suivante.

Nous avons également implémenté une fonction permettant la suppression des éléments tels que les bâtiments, salles, point d'accès et mesures de la base de données (cette fonction n'avait pas été implémentée par l'équipe précédente) afin d'éviter de se retrouver avec une liste déroulante énorme et une base de donnée toujours plus grande à chaque mesures. Cela permet également d'effacer un élément mal configuré (mauvaises coordonnées d'un bâtiment, mauvais ESSID d'un point d'accès,…).

Ces améliorations d'ergonomie permettent une utilisation plus simple et intuitive du logiciel par l'utilisateur.

II.4.6) Fiabilisation

II.4.6.1) Corrections de bugs

La correction de bugs a été une phase qui s'est étalée tout au long du projet. En effet, il nous a fallu premièrement corriger les bugs présents dans le programme développé l'année précédente puis les bugs apparus lors de notre propre développement.

II.4.6.2) Mise en place de tests

Nous avons également implémentés de nombreux tests dans le programme afin de sécuriser son fonctionnement.

- Tests de la validité des valeurs entrées par l'utilisateur
- Vérification du fonctionnement de la base de données avant de démarrer le programme

Une fois le programme terminé, nous avons également procédé à une batterie de test sur le programme afin de valider toutes les fonctions dans tous les cas de figure, vérifiant ainsi le bon fonctionnement et la stabilité du programme.

II.4.7) Distribution

Afin de faciliter la distribution de GWA, nous avons choisit de proposer 2 modes différents : le packaging Debian et la Clef USB.

II.4.7.1) Packaging

Dans le cadre du déploiement de GWA, nous avons créé un paquet debian pour simplifier son installation. Ce paquet permet de s'assurer que toutes les dépendances nécessaires au bon fonctionnement de GWA sont installées sur le système, d'installer le logiciel et de configurer la base de données. La finalité étant dans un premier temps d'intégrer le paquet sur un dépôt non officiel de manière à l'installer via apt et enfin, pour plus tard, de l'intégrer aux dépôts officiels debian/ubuntu. Cette dernière étape étant très compliqué car pour être accepté dans les dépôts officiels, le paquet doit respecter un certain nombre de règles très strictes.

Dans ce qui suit, nous allons détailler assez simplement comment régénérer le paquet, mais il faut savoir que le packaging debian est quelque chose de très sensible et pas forcément très simple a appréhender. En effet, il est possible de casser un système avec un paquet mal conçu. C'est pour ca qu'après avoir lu un peu cette partie, nous vous recommandons, si vous n'avez jamais pratiqué de packaging debian, d'aller consulter de la documentation sur ce point, notamment :

- http://www.debian.org/doc/maint-guide/ Ce document est LA référence du maintenant debian, le problème est qu'elle est très longue et parfois 'indigeste'
- http://doc.ubuntu-fr.org/tutoriel/creer_un_paquet Version plus facile à appréhender. Etant donné que cette doc est relativement courte, il est conseillé **de la lire en entier** avant même de commencer à générer un nouveau paquet pour GWA.

Fonctionnement du paquet

Le paquet va copier tout le programme dans /usr/share/python-support/gwa. Un script de lancement *gwa.sh* est copié dans /usr/bin tandis qu'un fichier gwa.desktop (fichier pour les menus gnomes/kde) est copié dans /usr/share/applications/.

Lors de la première installation, le paquet va importer les dumps dans la BDD postgresql. Cette étape n'est pas faite lors des mises à jour (a moins d'une grosse mise à jour au niveau de la base de données)

Génération du paquet

Paquets nécessaires pour la génération : cdbs devscripts debhelper build-essential fakeroot lintian

Pour commencer, récupérer le fichier compressé qui contient le squelette du paquet sur le site du projet et le décompresser dans votre répertoire de travail. L'arborescence est la suivante :

```
.
-- gwa-0.30
   |-- debian
   |-- dumpDB
   |-- images
   |-- python
```

Le dossier *debian* contient toutes les informations relatives au paquet (comment le construire, la description, les dépendances, …), *dumpDB* contient les dumps initiaux de la base de donnée, *images* contient les images nécessaires au fonctionnement du programme et enfin *python* contient le programme en lui-même.

Lors d'une régénération, il convient de copier les nouveaux fichiers au bon endroit (images dans images et les sources python dans python). On n'est pas obligé de faire le tri avec les fichiers .pyc ou autre car ils seront automatiquement supprimés lors la génération du paquet.

Une fois ces fichiers copiés, placez vous dans le dossier debian/ :

```
user@mercure:~/projet/packaging/GWA/gwa-0.20$ cd debian
```

Puis lancer la commande *dch -i*

```
user@mercure:~/projet/packaging/GWA/gwa-0.20/debian$ dch -i
```

Cette commande permet de modifier correctement le fichier changelog qui est la base lors d'un upgrade de version. Il va lancer un éditeur de texte sur ce fichier. Voici un exemple d'ajout :

```
gwa (0.30-1) jaunty; urgency=low
  * New dump From SVN
 -- Sébastien CANCHON <canchons@esiee.fr>  Tue, 06 Apr 2010 16:06:47 +0200
```

Pour indiquer le changement de version (majeure, tout changement de version après le tiret est considéré comme un changement au niveau du packaging et non du programme), dch renommera le dossier source:

```
dch warning: your current directory has been renamed to:
../gwa-0.30
```

On peut maintenant lancer la commande pour générer le paquet. Pour cela, il faut se mettre a la racine du paquet et lancer la commande 'debuild --no-lintian -sa':

```
user@mercure:~/projet/packaging/GWA/gwa-0.30$ debuild --no-lintian -sa
This package has a Debian revision number but there does not seem to be
an appropriate original tar file or .orig directory in the parent directory;
(expected gwa_0.30.orig.tar.gz or gwa-0.30.orig)
continue anyway? (y/n) y
 dpkg-buildpackage -rfakeroot -D -us -uc -sa
[…]
dpkg-deb : construction du paquet « gwa » dans « ../gwa_0.30-2_i386.deb ».
 dpkg-genchanges -sa >../gwa_0.30-2_i386.changes
dpkg-genchanges: inclusion du code source original dans l'envoi (« upload »)
dpkg-buildpackage: envoi complet ; paquet Debian natif (inclusion du code source complet)
Now signing changes and any dsc files...
 signfile gwa_0.30-2.dsc Sébastien CANCHON canchons@esiee.fr
gpg: « Sébastien CANCHON <canchons@esiee.fr> » a été ignoré: la clé secrète n'est pas
disponible
gpg: [stdin]: clearsign failed: la clé secrète n'est pas disponible
debsign: gpg error occurred!  Aborting....
debuild: fatal error at line 1250:
running debsign failed
```

On a une erreur ce qui est normal car nous n'avons pas de signature gpg officielle.

On vérifie ensuite que notre paquet "gwa_0.30-1_i386.deb" a bien été créé dans le répertoire parent de celui ou l'on est actuellement:

```
user@mercure:~/projet/packaging/GWA/gwa-0.30$ ls ..
gwa_0.30-1_i386.build     gwa_0.30-1_i386.changes
gwa-0.30                  gwa_0.30-1_i386.deb
gwa_0.30-1.dsc            gwa_0.30-1.tar.gz gwa_0.30-1.dsc.asc
```

Tous ces fichiers ont leur utilité dans le cas d'une intégration dans un dépôt. Pour nous, tant que nous n'avons pas de dépôt, seul le fichier deb nous intéresse.

II.4.7.2) Clef USB

Pour une utilisation très simple de GWA sur un pc portable et afin de ne pas être dépendant de l'OS installé sur ce pc, nous avons décidé de produire une clef USB bootable directement dans un environnement personnalisé pour GWA et contenant tous les programmes nécessaires à son utilisation et son développement futur :

- Kubuntu 9.10
- GWA
- PostgreSQL
- Librairie de développement Python
- PyQT4
- QT Designer 4
- Kate
- Vim
- Firefox
- Eric
- Subversion
- Inkscape
- Aircrack-ng
- Wireshark

L'avantage est de pouvoir utiliser rapidement GWA sans aucun impact sur l'OS installé sur le pc portable hôte.

Divers outils permettent de créer une clé USB avec une distribution linux bootable, mais aucune ne permet de la créer directement personnalisée. Cependant, la personnalisation du filesystem est possible. Il a donc été décidé de procéder de la manière suivante :

1. Création de la clé USB bootable avec un outil tel qu'UnetBootIn ou Linux Live USB Creator
2. Extraction et personnalisation du filesystem
3. Remplacement du filesystem de la clé USB

1) Le point 1 n'étant que suivre les indications d'une Interface Utilisateur, on ne détaillera pas cette partie.

2) Extraction et personnalisation du filesystem

Le filesystem d'une liveUSB (comme un liveCD) est contenu dans le fichier filesystem.squashfs.

```
Mount /dev/sdc1 /tmp/usb
```

Pour personnaliser le filesystem, il faut pouvoir le monter, pour l'explorer, le modifier et se chrooter dedans. Pour cela, on va le monter, et comme squashfs est un système de fichiers en lecture seule, on va copier son contenu afin de pouvoir le modifier.

```
Mount –o loop –t squashfs /tmp/usb/casper/filesystem.squashfs ~/squashfs/

Cp –aR ~/squashfs/ ~/custom/

Cd -

Cp /etc/hosts /etc/resolv.conf etc/

Chroot .
```

Nous voilà maintenant dans le filesystem qui sera contenu dans la clé. Toutes les opérations suivantes prendront effet non pas sur notre système de développement mais sur le futur filesystem de la clé USB. C'est ici que l'on va installer / supprimer des paquets, ou encore créer certains fichiers.

```
mount -t proc none /proc/

mount -t sysfs none /sys/

export HOME=/root

apt-get remove --purge liste des paquets à supprimer

apt-get install liste des paquets à installer

wget sources de GWA…

…
```

Avant de fermer le chroot dans l'état souhaité pour la clé USB, on va le nettoyer un peu.

```
Rm –rf /tmp/*

Apt-get clean

Rm –f /etc/hosts /etc/resolv.conf

Exit
```

Nous revoilà maintenant dans notre système de développement. Nous allons d'abord mettre à jour certains fichiers, puis recréer un nouveau squashfs, qui remplacera celui de la clé USB.

3) Remplacement du filesystem de la clé USB

```
chmod +w /tmp/usb/casper/filesystem.manifest
chroot ~/custom dpkg-query -W --showformat='${Package} ${Version}\n' > /tmp/usb/casper/filesystem.manifest
/tmp/usb/casper/filesystem.manifest /tmp/usb/casper/filesystem.manifest-desktop

mksquashfs ~/custom /tmp/usb/casper/filesystem.squashfs

Parallel mksquashfs: Using 4 processors

…
```

Maintenant, Nous allons mettre à jour deux derniers fichier : Le README (facultatif) situé à /mnt/usb/README, et le fichier de checksum de notre nouvelle distribution.

```
(cd /tmp/usb && find . -type f -print0 | xargs -0 md5sum > md5sum.txt)
```

Voilà il ne nous reste plus qu'à rebooter un PC sur notre clé USB

II.4.8) Documentation du projet

II.4.8.1) Documentation du code

Afin de faciliter la reprise du projet par les prochaines équipes, nous avons commenté en détail le code Python. Nous avons également généré une documentation HTML à partir des commentaires présents dans le code.

Doxygen est un logiciel libre de génération de documentation de code. Ce dernier peut être utilisé dans nombre de langages (JAVA, C, C++, …) et notamment pour Python en lui ajoutant un filtre correct. Il repose sur un système de commentaires et de docstring très similaires à ceux utilisés dans la JAVADOC. Exemple de commentaires :

```
“““
Fonction toto
@param titi parametre 1
@param tutu parametre2
“““
```

Il est possible d’avoir plus de détails sur les docstrings spécifiques à python sur le site officiel de doxygen : http://www.doxygen.nl/docblocks.html#pythonblocks

Le fonctionnement de doxygen est basé sur la lecture d’un fichier de configuration propre au projet. Ce fichier utilise des chemins relatifs et est donc assez sensible. Pour éviter devoir réécrire complément le fichier de config, assurez vous d’avoir récupérer correctement notre dépôt SVN et d’avoir une arborescence de ce type :

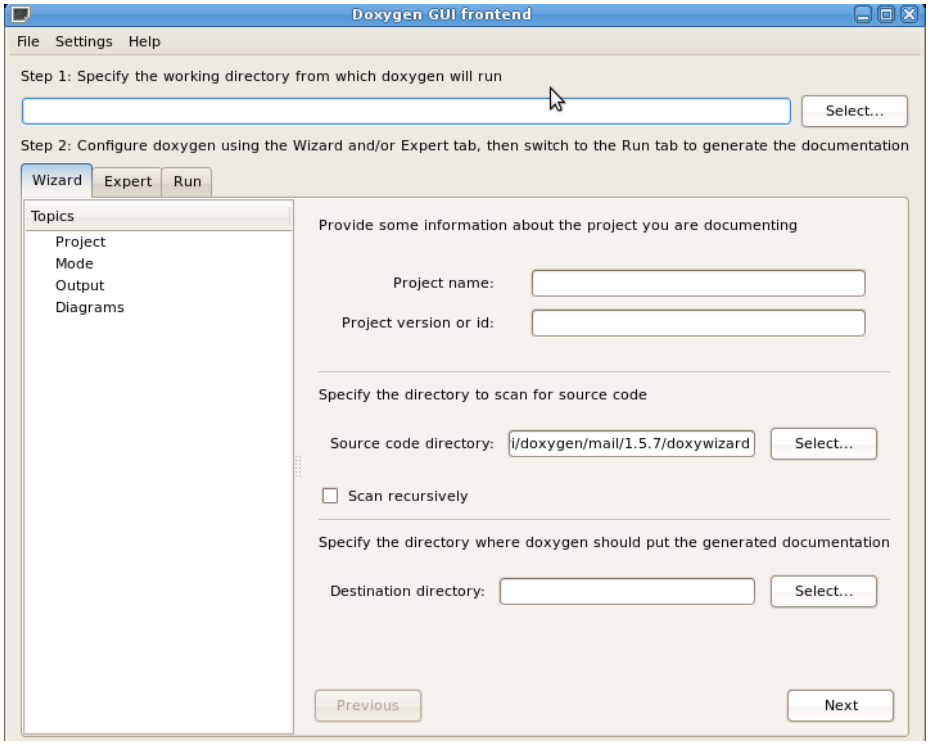
```
.
|-- DOC
|-- Projet_final
|   |-- New_pagesui
|   |-- images
|   |-- python
|   |   |-- DAC
|   |   |   `-- DBA
|   |   |       `-- Mapping
|   |   |           `-- Files
|   |   `-- PythonUI
|   |-- scripts
```

```
|   `-- tools
`-- dumpBD
```

Dans notre cas de fichier est nommé ***doxygen_gwa*** et se trouve dans le répertoire « *Projet_Final/scripts/* » du SVN. Ce fichier a été configuré par nos soins pour générer des fichiers *LaTeX* dans le répertoire DOC/. Ces fichiers latex peuvent ensuite être convertis en PDF.

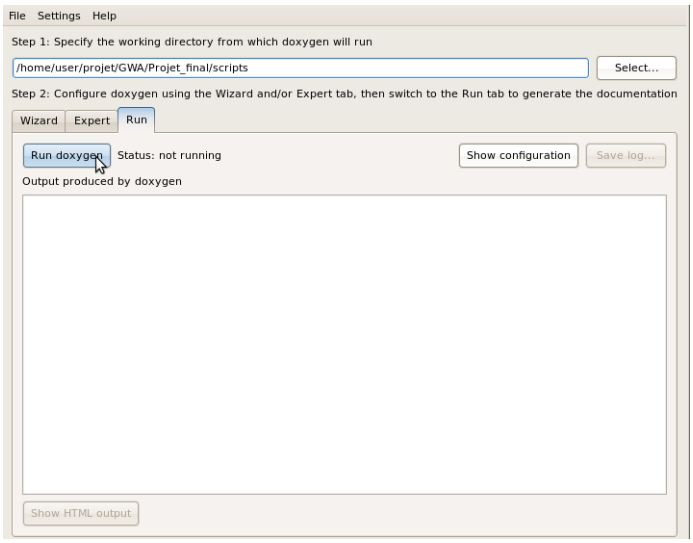
Génération de la documentation :

- ***Première étape :*** Pour que doxygen interprète correctement le python, il est nécessaire d'installer un filtre supplémentaire. Ce filtre s'appel **doxypy** et il est possible de trouver les informations pour le télécharger et l'installer (au moment de la rédaction de ce rapport) ici : http://code.foosel.org/doxypy. Ce filtre est nécessaire pour la génération de la documentation de GWA avec notre fichier de configuration doxygen
- ***Seconde étape :*** il faut installer les paquets *doxygen* et *doxywizard*. Le premier fourni le logiciel doxygen en lui-même, tandis que le deuxième fourni un front-end graphique pour l'utilisation de doxygen (qui a la base s'utilise uniquement en ligne de commande).
- ***Troisième étape :*** exécuter la commande *doxywizard*. Vous devriez avoir une interface qui ressemble à cette capture d'écran :

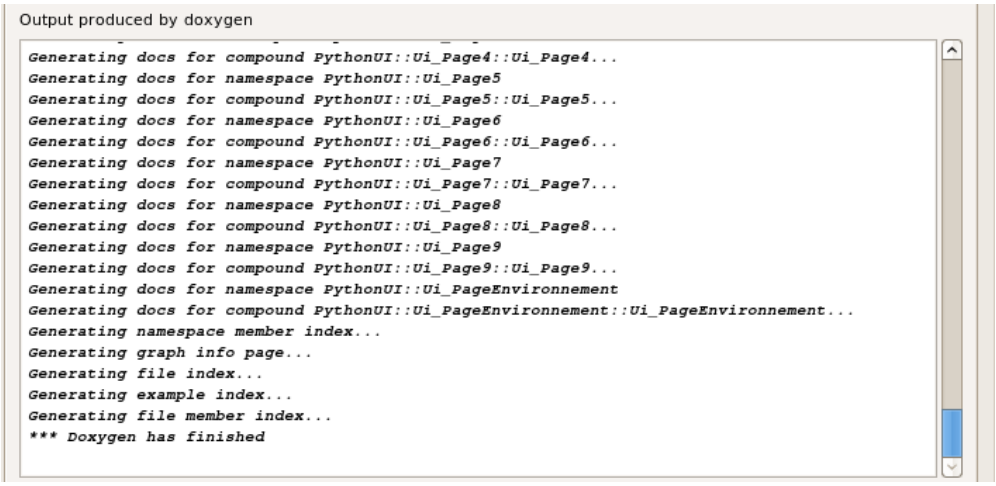


L'interface de doxywizard est assez épurée. Trois onglets sont disponibles : deux consacrés au fichier de configuration (un pour en mode simplifié qui se nomme Wizard et un deuxième avec plus de détails nommé Expert) et un troisième pour lancer doxygen pour générer la documentation.

- ***Quatrième étape :*** Il faut charger le fichier de configuration de doxygen pour GWA via File/Open dans doxywizard. Une fois chargé, vous pouvez regarder un peu quels sont les options qui différent par rapport à une configuration par défaut : ces options sont marquées en rouge dans l'onglet expert.
- **Cinquième étape :** on va maintenant générer la documentation. Rendez vous dans l'onglet 'Run' puis cliquez sur 'Run Doxygen'.



La sortie de la commande doxygen s'affichera au milieu de l'onglet :


Output produced by doxygen
Generating docs for compound PythonUI::Ui_Page4::Ui_Page4...
Generating docs for namespace PythonUI::Ui_Page5
Generating docs for compound PythonUI::Ui_Page5::Ui_Page5...
Generating docs for namespace PythonUI::Ui_Page6
Generating docs for compound PythonUI::Ui_Page6::Ui_Page6...
Generating docs for namespace PythonUI::Ui_Page7
Generating docs for compound PythonUI::Ui_Page7::Ui_Page7...
Generating docs for namespace PythonUI::Ui_Page8
Generating docs for compound PythonUI::Ui_Page8::Ui_Page8...
Generating docs for namespace PythonUI::Ui_Page9
Generating docs for compound PythonUI::Ui_Page9::Ui_Page9...
Generating docs for namespace PythonUI::Ui_PageEnvironnement
Generating docs for compound PythonUI::Ui_PageEnvironnement::Ui_PageEnvironnement...
Generating namespace member index...
Generating graph info page...
Generating file index...
Generating example index...
Generating file member index...
*** Doxygen has finished

- ***Sixieme etape :*** Une fois la génération finie, il faut aller dans le répertoire « DOC/latex/ » puis lancer la commande **make**. Cette commande s'occupe de combiner toutes les pages LaTeX de manière à produire un seul et unique pdf qui sera nommé *refman.pdf*

II.4.8.2) Etude concurrentielle

Afin de pouvoir cerner les avantages de GWA, nous avons réalisé une rapide étude concurrentielle afin d'identifier quels sont les produits semblables à GWA sur le marché. Nous avons ainsi pu voir les avantages et inconvénients de chacun et remarqué que le segment visé par le logiciel GWA était encore libre.

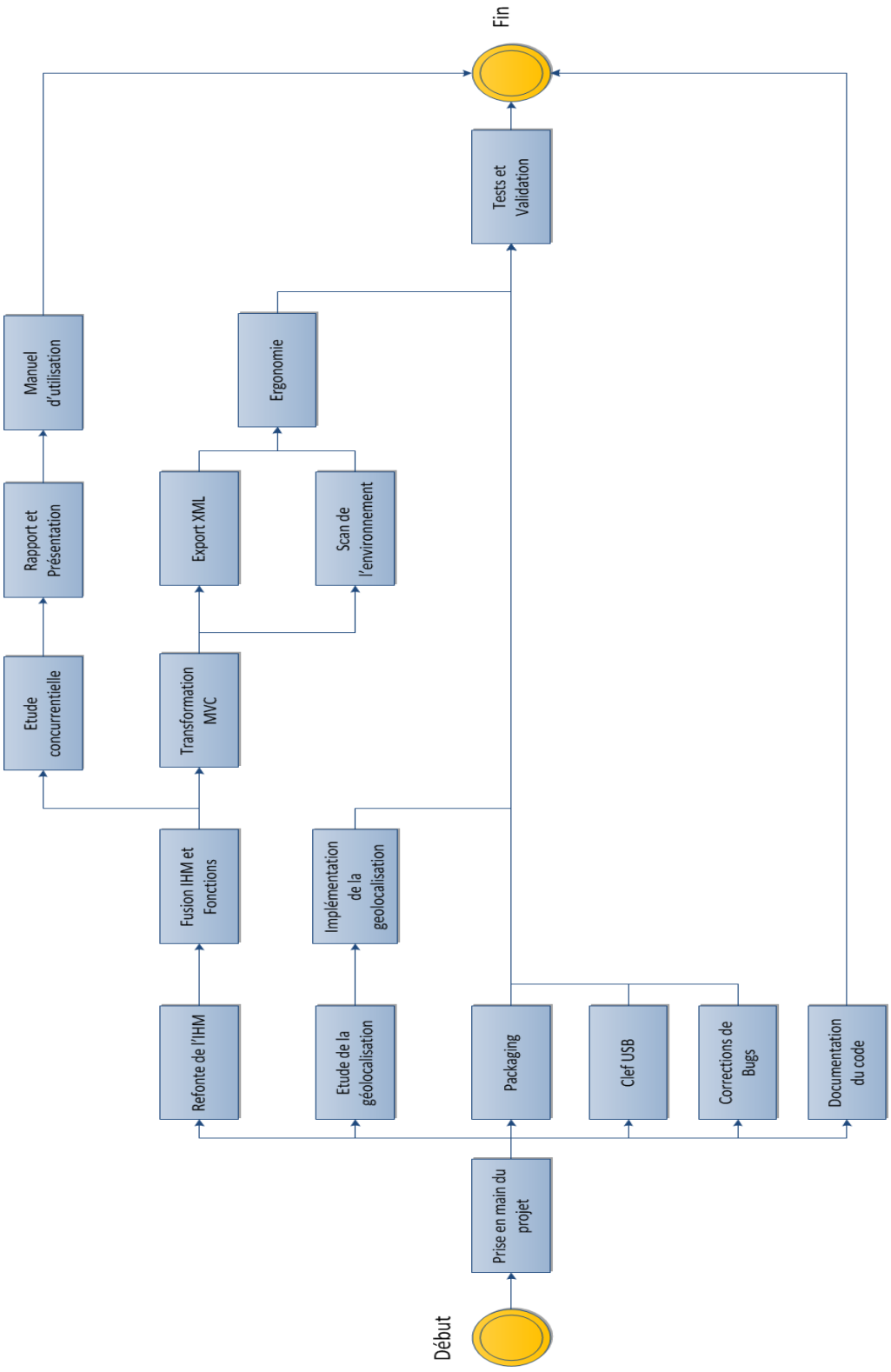
II.4.8.3) Rapport et présentation

A partir de la seconde période du projet, nous avons démarré la rédaction du rapport de projet ainsi que la préparation de la soutenance.

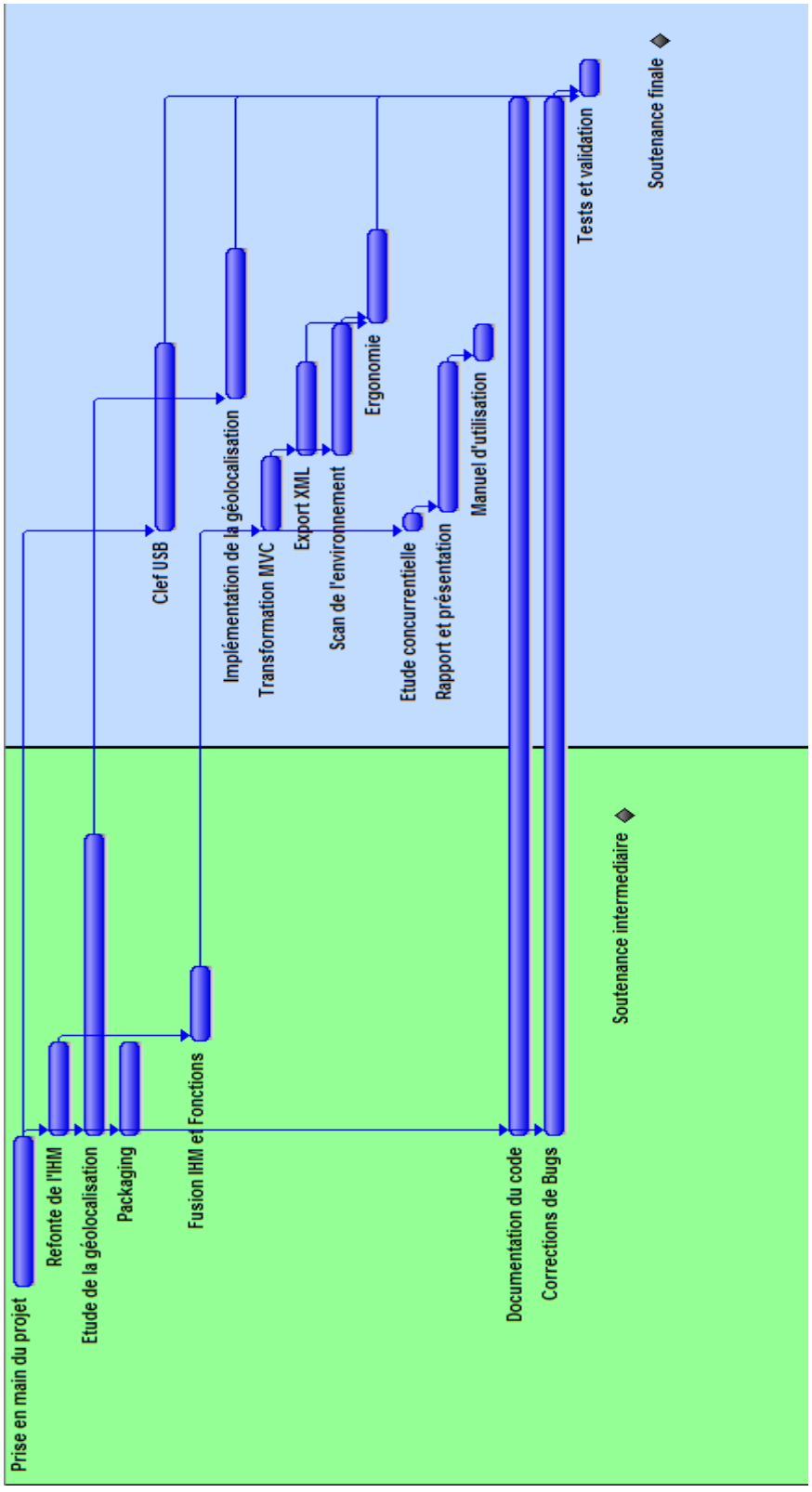
II.4.8.4.) Manuel du programme

Afin d'expliquer le fonctionnement du logiciel aux utilisateurs, nous avons rédigé un manuel d'utilisation. Ce manuel comprend des captures d'écran de chaque étape et permet d'expliquer à l'utilisateur comment remplir les différents champs. Ce document est fournit en annexe IV.2.

II.5) Diagramme de PERT



II.6) Diagramme de GANTT



Le diagramme de GANTT est donné à titre indicatif. Les tâches au sein du carré vert sont celles effectuées lors de la première période. Les tâches dans le carré bleu sont celles effectués dans la seconde période.

II.7) Indicateurs

Afin de pouvoir suivre l'évolution du projet, nous avons mis en place des points d'avancements réguliers avec les personnes de l'équipe. Ces points nous ont permit d'échanger sur l'avancement des tâches de chacun et de pouvoir s'entraider et discuter des problèmes à résoudre. Nous avons également mis en place une liste de tâches sur le Wiki du Google Code afin renseignant les tâches, les personnes attribuées à ces tâches et l'état d'avancement.

Nous avons également effectué un bilan intermédiaire correspondant à la moitié du projet, c'est-à-dire à la fin de la première période (20 heures de projet). Ce bilan intermédiaire nous a permis de faire un état complet d'avancement et de préparer la seconde période. Cela nous a permis d'établir la liste suivante :

- Tâches accomplies
- Tâches restant à accomplir
- Problématiques à résoudre et difficultés

Nous avons également fait une soutenance avec le commanditaire notamment afin de rendre compte de l'avancement de ce projet.

II.8) Problèmes rencontrés

II.8.1) Planification

- Planification des horaires trop dense :
  - Perte de nombreuses heures lors d'un blocage pour lequel on doit attendre une intervention extérieure
- Planification de l'intervention des consultants inadaptée :
  - Les interventions de Sébastien Panel sont arrivées trop tard (en seconde période) alors que son aide aurait été utile au démarrage du projet afin de nous aider à nous l'approprier.
  - Les interventions d'Olivier Cayrol sont arrivées trop tôt (dès le départ du projet) au moment ou nous tentions de nous approprier le projet. Son aide en Python a été utile au moment ou nous avons commencé le développement.

II.8.2) Techniques

- Démarrage du projet sans connaitre le langage Python
- Dimensions des fenêtres rendant l'ancien projet inutilisable
- Sources de l'IHM non fournies au départ
- Fichiers IHM ne correspondant plus au programme, codage de l'interface directement dans les sources Python par l'ancienne équipe
- Fonctions mathématiques

III. Conclusion

III.1) Conclusion technique et humaine

Ce projet nous a permis de réaliser vrai projet de développement logiciel. Nous avons ainsi pu améliorer nos compétences en gestion de projet (gestion des plannings, répartition des tâches) et découvrir des outils de développement collaboratifs (SVN, plateforme GoogleCode). Nous avons également amélioré nos compétences en langage Python.

Humainement, ce projet nous a permis de découvrir le travail en équipe et de voir l'importance de la communication entre les membres pour la bonne réalisation du projet.

Nous sommes heureux d'avoir pu contribuer au développement d'un logiciel open-source amené à évoluer à l'ESIEE. Il peut dors et déjà être utilisé par tout un chacun afin d'établir la couverture Wi-Fi de son domicile. Il se positionne sur un segment de marché totalement libre et pourrait donc d'ici quelques années devenir un logiciel très populaire.

III.2) Axes de développement futurs

GWA étant destiné à poursuivre son développement les années suivantes, des axes de développements peuvent être dessinés pour les futures versions :

Carte thermique du débit et mix puissance/débit

Les mesures de puissance Wi-Fi sont représentées à la fin de la campagne sous forme de carte thermique. Il serait intéressant par la suite d'implémenter la même chose pour les mesures de débit ainsi que produire une cartographie mêlant les données de couverture et de débit.

Implémentation complète de l'étude de géolocalisation

Nous avons pu réaliser l'étude mathématique complète permettant de replacer les mesures des salles au sein du bâtiment. Cependant, nous n'avons pu implémenter que des exemples en Python. Pour les prochaines versions, il serait intéressant de pouvoir implémenter totalement l'étude mathématique afin d'ajouter cette fonction au programme.

Amélioration de l'IHM

Nous avons modifié l'IHM créer l'année précédente afin de l'alléger et de la rendre plus ergonomique. Il serait intéressant de pouvoir améliorer encore l'interface en la rendant encore plus simple et graphique, à la manière d'un assistant Windows.

Intégration des fichiers SVG

Nous n'avons pas pu intégrer la gestion des fichiers SVG. Les plans utilisés par le logiciel doivent être sous forme JPEG ou PNG. Il serait très intéressant de pouvoir implémenter le support de ce format avec la possibilité d'ouvrir et de sauvegarder les plans sous ce format.

IV. Annexes

IV.1) Statistiques

- Entre 80 et 90 heures de travail réparties sur 2 périodes d'apprentissage
- 256 révisions
- 92 fichiers
- 3113 lignes de code Python
    - 4600 pour le projet initial
- 10 226 lignes de codes modifiées
    - Ajout / Suppression / Modification
- 23 Bugs ou améliorations signalés
    - 20 Bugs ou améliorations corrigés
- Paquet Debian de 333ko pour la release 0.3 (stable)

IV.2) Manuel d'utilisation

La manuel d'utilisation est fournit avec le projet. Il se nomme « GWA - Manuel d'utilisation.pdf »